



TUGAS AKHIR - KI141502

# MODIFIKASI ALGORITMA *CONTROLLING EXPANSION* UNTUK STEGANOGRAFI PADA CITRA

HERDITO IBNU DEWANGKORO  
NRP 05111440000023

Dosen Pembimbing I  
Tohari Ahmad, S.Kom., MIT., Ph.D.

Dosen Pembimbing II  
Prof. Ir. Supeno Djanali, M.Sc., Ph.D.

Departemen Informatika  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018









**TUGAS AKHIR - KI141502**

# **MODIFIKASI ALGORITMA *CONTROLLING EXPANSION* UNTUK STEGANOGRAFI PADA CITRA**

**HERDITO IBNU DEWANGKORO  
NRP 05111440000023**

**Dosen Pembimbing I  
Tohari Ahmad, S.Kom., MIT., Ph.D.**

**Dosen Pembimbing II  
Prof. Ir. Supeno Djanali, M.Sc., Ph.D.**

**Departemen Informatika  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018**

*[Halaman ini sengaja dikosongkan]*



**UNDERGRADUATE THESES - KI141502**

# **MODIFIED CONTROLLING EXPANSION ALGORITHM FOR IMAGE STEGANOGRAPHY**

**HERDITO IBNU DEWANGKORO  
NRP 05111440000023**

**First Advisor**

**Tohari Ahmad, S.Kom., MIT., Ph.D.**

**Second Advisor**

**Prof. Ir. Supeno Djanali, M.Sc., Ph.D.**

**Department of Informatics**

**Faculty of Information and Communication Technology**

**Sepuluh Nopember Institute of Technology**

**Surabaya 2018**

*[Halaman ini sengaja dikosongkan]*



## LEMBAR PENGESAHAN

### MODIFIKASI ALGORITMA *CONTROLLING EXPANSION* UNTUK STEGANOGRAFI PADA CITRA

#### TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada  
Bidang Studi Komputasi Berbasis Jaringan  
Program Studi S-1 Departemen Informatika  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember

Oleh:

**HERDITO IBNU DEWANGKORO**

**NRP: 05111440000023**

Disetujui oleh Pembimbing Tugas Akhir:

1. Tohari Ahmad, S.Kom., M.T., Ph.D .....  
(NIP. 197505252003121002) (Pembimbing 1)
2. Prof. Ir. Supeno Djanali, M.Sc., Ph.D .....  
(NIP. 194806191973011001) (Pembimbing 2)

**SURABAYA  
JULI, 2018**

*[Halaman ini sengaja dikosongkan]*

# **MODIFIKASI ALGORITMA *CONTROLLING EXPANSION* UNTUK STEGANOGRAFI PADA CITRA**

**Nama Mahasiswa : HERDITO IBNU DEWANGKORO**  
**NRP : 05111440000023**  
**Departemen : Informatika FTIK-ITS**  
**Dosen Pembimbing 1 : Tohari Ahmad, S.Kom., MIT., Ph.D.**  
**Dosen Pembimbing 2 : Prof. Ir. Supeno Djanali, M.Sc.,  
Ph.D.**

## **Abstrak**

*Steganografi merupakan suatu teknik untuk menyembunyikan data ke dalam media steganografi sehingga hanya pengirim dan penerima media steganografi saja yang dapat mengetahui data rahasia yang disisipkan. Citra merupakan salah satu media yang dapat digunakan untuk steganografi. Salah satu metode steganografi yang menggunakan media citra adalah Controlling Expansion.*

*Controlling Expansion menggunakan satu pixel pada cover image untuk menyisipkan suatu payload dengan bentuk bilangan basis 2. Pada Tugas Akhir ini akan dilakukan modifikasi dari algoritma Controlling Expansion. Modifikasi dilakukan dengan memaksimalkan payload yang disisipkan sampai dengan bilangan basis 5. Metode hasil modifikasi diharapkan dapat meningkatkan kapasitas data yang dapat disisipkan pada cover image dengan memaksimalkan kualitas stego image.*

*Pengujian dilakukan menggunakan tiga macam grayscale image berukuran 8-bit. Data yang disisipkan berupa daftar payload dengan jenis bilangan basis 2 dengan ukuran 10 kb, 50 kb, dan 100 kb yang juga akan dikonversikan sampai dengan bilangan basis 5. Data yang didapat dari hasil pengujian menunjukkan bahwa modifikasi metode Controlling Expansion yang dilakukan dapat meningkatkan kapasitas payload yang disisipkan.*

***Kata kunci: Steganografi, Data hiding, Citra, Controlling Expansion.***

## **MODIFIED CONTROLLING EXPANSION ALGORITHM FOR IMAGE STEGANOGRAPHY**

**Student's Name : HERDITO IBNU DEWANGKORO**  
**Student's ID : 0511140000023**  
**Department : Informatika FTIK-ITS**  
**First Advisor : Tohari Ahmad, S.Kom., MIT., Ph.D.**  
**Second Advisor : Prof. Ir. Supeno Djanali, M.Sc., Ph.D.**

### ***Abstract***

*Steganography is a technique to hide data into steganographic media so that only the sender and receiver of steganographic media can know the inserted secret data. Image is one of the media that can be used for steganography. One of the steganographic method that using image as a steganographic media is Controlling Expansion.*

*Controlling Expansion uses one pixel in the cover image to insert payloads with the base number 2. This undergraduate thesis will be modifying the Controlling Expansion algorithm. Modification is done by maximizing the payloads that are inserted up to the base number 5. The modified method is expected to increase the capacity of data that can be inserted to the cover image by maximizing the quality of stego image.*

*The test was performed using three kinds of 8-bit grayscale image. The data inserted in the form of the list of payloads with the type of base number 2 with the size of 10 kb, 50 kb, and 100 kb which will also be converted up to the base number 5. Data that obtained from the test results indicate that the modification of Controlling Expansion method can be increasing the payload capacity that can be inserted.*

***Keywords: Steganography, Data hiding, Image, Controlling Expansion.***

***[Halaman ini sengaja dikosongkan]***

## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

*Assalamu 'alaikum warahmatullahi wabarakatuh*

*Alhamdulillah Rabbil 'alamin*, segala puji bagi Allah *Subhanahu wa Ta'ala*, Tuhan yang Mahasuci dan Mahatinggi, yang telah memberikan hidayah, nikmat, dan kekuatan-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul:

### **“MODIFIKASI ALGORITMA *CONTROLLING* *EXPANSION* UNTUK STEGANOGRAFI PADA CITRA”.**

Selawat serta salam senantiasa tercurah kepada Rasulullah Muhammad *Shallallahu 'Alaihi wa Sallam* yang telah menjadi sebaik-baik teladan bagi umatnya dalam segala hal.

Terselesaikannya pengerjaan Tugas Akhir ini tentunya tidak terlepas dari dukungan berbagai pihak, sehingga pada kesempatan ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Ibu dan Bapak penulis yang tidak kenal lelah dalam memberikan begitu banyak nasihat, arahan, dukungan, dan motivasi sehingga penulis dapat menyelesaikan Tugas Akhir ini.
2. Kakak penulis dan istri, yang telah memberikan bimbingan dan motivasi kepada penulis.
3. Adik penulis, yang telah memberikan hiburan dan semangat kepada penulis.
4. Bapak Tohari Ahmad, S.Kom., MIT., Ph.D. selaku dosen wali dan dosen pembimbing 1 penulis yang telah memberikan dukungan, bimbingan, dan motivasi untuk menyelesaikan Tugas Akhir ini.

5. Bapak Prof. Ir. Supeno Djanali, S.Kom., M.Sc., Ph.D. selaku dosen pembimbing 2 penulis yang telah memberikan dukungan, bimbingan, dan motivasi untuk menyelesaikan Tugas Akhir ini.
6. Bapak Dr.Eng. Darlis Herumurti, S.Kom., M.Kom. selaku Kepala Departemen Informatika ITS yang telah berusaha untuk memberikan yang terbaik selama penulis duduk di bangku perkuliahan.
7. Bapak Imam Kuswardayan S.Kom., M.T. selaku Pembina Lembaga Dakwah Departemen Keluarga Muslim Informatika (KMI) ITS yang telah memberikan ilmu, nasihat, motivasi, dan arahan kepada penulis, terutama ketika penulis menjadi Anggota KMI ITS.
8. Mas Ade Ilham Fajri, S.Kom. yang telah memberikan banyak nasihat, arahan, serta tidak bosan-bosannya untuk menjadi tempat berbagi ilmu bagi penulis.
9. Rekan-rekan Anggota KMI ITS Ramah Bertauhid dan Sahabat Bersinergi yang telah membagikan berbagai pelajaran dan pengalaman.
10. Rekan-rekan Rumpun Mata Kuliah (RMK) Komputasi Berbasis Jaringan (KBJ) yang telah membarengi penulis dan bersedia menjadi tempat dalam berdiskusi.
11. Rekan-rekan Administrator Laboratorium KBJ yang telah memberikan *support* kepada penulis.
12. Rekan-rekan Informatika ITS angkatan 2014, yang telah memberikan dukungan serta mewarnai kehidupan penulis selama menjalani perkuliahan.
13. Berbagai pihak yang tidak dapat penulis sebutkan satu per satu yang telah membantu penulis selama berkuliah di Informatika ITS.

Penulis menyadari bahwa Tugas Akhir ini masih terdapat banyak kekurangan sehingga dengan segala kerendahan hati penulis mengharapkan kritik dan saran untuk perbaikan ke depan. Akhir kata, penulis berharap bahwa Tugas Akhir ini dapat



bermanfaat bagi berbagai pihak yang membutuhkan. Semoga Allah *Subhanahu wa Ta'ala* membalas seluruh kebaikan yang telah dilakukan.

*Wassalamu 'alaikum Warahmatullahi Wabarakatuh*

Surabaya, Juli 2018

Penulis

*[Halaman ini sengaja dikosongkan]*

## DAFTAR ISI

<b>LEMBAR PENGESAHAN.....</b>	<b>v</b>
<b>Abstrak.....</b>	<b>vii</b>
<b>Abstract .....</b>	<b>ix</b>
<b>KATA PENGANTAR .....</b>	<b>xi</b>
<b>DAFTAR ISI.....</b>	<b>xv</b>
<b>DAFTAR GAMBAR.....</b>	<b>xix</b>
<b>DAFTAR TABEL.....</b>	<b>xxi</b>
<b>DAFTAR KODE SUMBER .....</b>	<b>xxiii</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	1
1.3 Batasan Permasalahan .....	2
1.4 Tujuan .....	2
1.5 Manfaat.....	2
1.6 Metodologi .....	2
1.6.1 Penyusunan Proposal Tugas Akhir .....	3
1.6.2 Studi Literatur .....	3
1.6.3 Implementasi Sistem.....	3
1.6.4 Pengujian dan Evaluasi.....	3
1.6.5 Penyusunan Laporan Tugas Akhir.....	4
1.7 Sistematika Penulisan Laporan .....	4
<b>BAB II TINJAUAN PUSTAKA .....</b>	<b>7</b>
2.1 Steganografi .....	7
2.2 <i>Difference Expansion</i> .....	7
2.3 <i>Reduced Difference Expansion</i> .....	8
2.4 <i>Controlling Expansion</i> .....	10
2.5 Python .....	12
2.6 Anaconda.....	12
2.7 OpenCV.....	13
2.8 <i>Mean Squared Error</i> .....	13
2.9 <i>Peak Signal-to-Noise Ratio</i> .....	14
2.10 <i>Correlation</i> .....	14
2.11 scikit-image .....	14

2.12 NumPy.....	15
2.13 Tkinter .....	15
<b>BAB III PERANCANGAN PERANGKAT LUNAK.....</b>	<b>17</b>
3.1 Data .....	17
3.1.1 Data Masukan .....	17
3.1.2 Data Keluaran .....	17
3.2 Deskripsi Umum .....	17
3.3 Perancangan Penyisipan <i>Payload</i> .....	19
3.4 Modifikasi Algoritma <i>Controlling Expansion</i> .....	20
3.5 Perancangan Antarmuka .....	26
<b>BAB IV IMPLEMENTASI.....</b>	<b>29</b>
4.1 Lingkungan Implementasi.....	29
4.2 Implementasi .....	29
4.2.1 Implementasi Pembacaan Citra.....	29
4.2.2 Implementasi Pengukuran Waktu Penyisipan/Ekstraksi <i>Payload</i> .....	30
4.2.3 Implementasi Penentuan Nilai <i>Random</i> .....	30
4.2.4 Implementasi Perhitungan <i>Correlation</i> .....	30
4.2.5 Modul <i>ModifiedControllingExpansion</i> .....	31
4.2.5.1 Implementasi Fungsi <i>__init__</i> .....	32
4.2.5.2 Implementasi Fungsi <i>get_difference</i> .....	33
4.2.5.3 Implementasi Fungsi <i>get_average</i> .....	33
4.2.5.4 Implementasi Fungsi <i>get_reduction</i> .....	34
4.2.5.5 Implementasi Fungsi <i>get_embed</i> .....	34
4.2.5.6 Implementasi Fungsi <i>get_pixel</i> .....	35
4.2.5.7 Implementasi Fungsi <i>run</i> .....	37
4.2.6 Modul <i>CalcPSNR</i> .....	38
4.2.7 Modul <i>Index</i> .....	38
4.2.7.1 Implementasi Fungsi <i>base_to_decimal</i> .....	39
4.2.7.2 Implementasi Fungsi <i>decimal_to_base</i> .....	40
4.2.7.3 Implementasi Fungsi <i>map_lm</i> .....	40
4.2.7.4 Implementasi Fungsi <i>restore_payload</i> .....	41
<b>BAB V HASIL UJI COBA DAN EVALUASI .....</b>	<b>43</b>
5.1 Lingkungan Pengujian.....	43
5.2 Data Uji .....	43

5.2.1 <i>Cover Image</i> .....	43
5.2.2 <i>Payload</i> .....	44
5.3 Skenario Uji Coba .....	44
5.3.1 Skenario Uji Coba 1 .....	44
5.3.2 Skenario Uji Coba 2 .....	48
5.3.3 Skenario Uji Coba 3 .....	51
5.4 Evaluasi Skenario Uji Coba .....	54
<b>BAB VI KESIMPULAN DAN SARAN .....</b>	<b>57</b>
6.1 Kesimpulan .....	57
6.2 Saran .....	57
<b>DAFTAR PUSTAKA .....</b>	<b>59</b>
<b>LAMPIRAN .....</b>	<b>63</b>
Lampiran A Kumpulan <i>Cover Image</i> .....	63
Lampiran B Kumpulan <i>Stego Image</i> .....	64
<b>BIODATA PENULIS .....</b>	<b>71</b>

*[Halaman ini sengaja dikosongkan]*

## DAFTAR GAMBAR

Gambar 2.1 Logo Python [9].....	12
Gambar 2.2 Logo Anaconda [11].....	13
Gambar 2.3 Logo OpenCV [13].....	13
Gambar 2.4 Logo scikit-image [18] .....	15
Gambar 2.5 Logo Numpy [20].....	15
Gambar 2.6 Logo Tkinter [22] .....	16
Gambar 3.1 Diagram Alir Gambaran Umum Modifikasi Algoritma <i>Controlling Expansion</i> .....	18
Gambar 3.2 Diagram Alir Penyisipan <i>Payload</i> .....	24
Gambar 3.3 Diagram Alir Ekstraksi <i>Payload</i> .....	25
Gambar 3.4 Desain Antarmuka .....	26

*[Halaman ini sengaja dikosongkan]*



## DAFTAR TABEL

Tabel 4.1 Lingkungan Implementasi Perangkat Lunak.....	29
Tabel 4.2 Daftar Fungsi Modul <i>ModifiedControllingExpansion</i>	32
Tabel 4.3 Daftar Fungsi Modul <i>Index</i> .....	39
Tabel 5.1 Lingkungan Pengujian.....	43
Tabel 5.2 Jumlah Digit <i>Payload</i> .....	44
Tabel 5.3 Nilai <i>PSNR</i> 10 kb <i>Payload</i> .....	45
Tabel 5.4 Nilai <i>Correlation (Cover Image &amp; Stego Image)</i> 10 kb <i>Payload</i> .....	46
Tabel 5.5 Waktu <i>Embedding</i> 10 kb <i>Payload</i> .....	46
Tabel 5.6 Waktu <i>Extracting</i> 10 kb <i>Payload</i> .....	47
Tabel 5.7 Nilai <i>Correlation (Payload &amp; Recovered Payload)</i> 10 kb <i>Payload</i> .....	47
Tabel 5.8 Nilai <i>Correlation (Cover Image &amp; Recovered Image)</i> 10 kb <i>Payload</i> .....	48
Tabel 5.9 Nilai <i>PSNR</i> 50 kb <i>Payload</i> .....	48
Tabel 5.10 Nilai <i>Correlation (Cover Image &amp; Stego Image)</i> 50 kb <i>Payload</i> .....	49
Tabel 5.11 Waktu <i>Embedding</i> 50 kb <i>Payload</i> .....	49
Tabel 5.12 Waktu <i>Extracting</i> 50 kb <i>Payload</i> .....	50
Tabel 5.13 Nilai <i>Correlation (Payload &amp; Recovered Payload)</i> 50 kb <i>Payload</i> .....	50
Tabel 5.14 Nilai <i>Correlation (Cover Image &amp; Recovered Image)</i> 50 kb <i>Payload</i> .....	51
Tabel 5.15 Nilai <i>PSNR</i> 100 kb <i>Payload</i> .....	51
Tabel 5.16 Nilai <i>Correlation (Cover Image &amp; Stego Image)</i> 100 kb <i>Payload</i> .....	52
Tabel 5.17 Waktu <i>Embedding</i> 100 kb <i>Payload</i> .....	52
Tabel 5.18 Waktu <i>Extracting</i> 100 kb <i>Payload</i> .....	53
Tabel 5.19 Nilai <i>Correlation (Payload &amp; Recovered Payload)</i> 100 kb <i>Payload</i> .....	53
Tabel 5.20 Nilai <i>Correlation (Cover Image &amp; Recovered Image)</i> 100 kb <i>Payload</i> .....	54

*[Halaman ini sengaja dikosongkan]*

## DAFTAR KODE SUMBER

<i>Pseudocode 4.1 Impelementasi Pembacaan Berkas Citra Grayscale 8-bit dengan Pustaka OpenCV .....</i>	30
<i>Pseudocode 4.2 Implementasi Pengukuran Waktu Penyisipan/Ekstraksi Payload .....</i>	30
<i>Pseudocode 4.3 Implementasi Penentuan Nilai Random .....</i>	30
<i>Pseudocode 4.4 Impelementasi Perhitungan Correlation .....</i>	31
<i>Pseudocode 4.5 Fungsi __init__ .....</i>	33
<i>Pseudocode 4.6 Fungsi get_difference .....</i>	33
<i>Pseudocode 4.7 Fungsi get_average .....</i>	34
<i>Pseudocode 4.8 Fungsi get_reduction.....</i>	34
<i>Pseudocode 4.9 Fungsi get_embed .....</i>	35
<i>Pseudocode 4.10 Fungsi get_pixel 1 .....</i>	36
<i>Pseudocode 4.11 Fungsi get_pixel 2 .....</i>	37
<i>Pseudocode 4.12 Fungsi run .....</i>	38
<i>Pseudocode 4.13 Implementasi Perhitungan PSNR dan MSE....</i>	38
<i>Pseudocode 4.14 Fungsi base_to_decimal.....</i>	39
<i>Pseudocode 4.15 Fungsi decimal_to_base .....</i>	40
<i>Pseudocode 4.16 Fungsi map_lm.....</i>	41
<i>Pseudocode 4.17 Fungsi restore_payload.....</i>	42

*[Halaman ini sengaja dikosongkan]*

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Dewasa ini perkembangan teknologi semakin pesat. Dengan pesatnya perkembangan teknologi tersebut, kebutuhan akan pertukaran data juga semakin dibutuhkan. Selain itu, keamanan data juga menjadi suatu perhatian untuk menghindari kerugian yang diakibatkan dari pencurian dan penyalahgunaan data oleh berbagai pihak yang tidak bertanggung jawab.

Steganografi merupakan teknik penyembunyian data memanfaatkan media tertentu dengan harapan pihak yang tidak memiliki hak akses atas data tersebut tidak dapat mengetahui bahwa ada data yang disembunyikan pada media steganografi [1]. Dengan demikian, steganografi dapat menjadi salah satu solusi untuk melakukan proteksi data. Terdapat berbagai macam media yang dapat digunakan pada steganografi, seperti media teks, citra, audio, dan video.

Saat ini pengembangan metode steganografi masih banyak dilakukan. Salah satunya pengembangan dilakukan untuk mengoptimalkan kapasitas penyisipan data. Dalam Tugas Akhir ini akan dilakukan penelitian untuk meningkatkan kapasitas penyisipan data serta mendapatkan kualitas *stego image* yang optimal pada steganografi dengan media citra menggunakan modifikasi algoritma *Controlling Expansion* [2].

### **1.2 Rumusan Masalah**

Rumusan masalah yang diangkat dalam Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana cara meningkatkan kapasitas penyisipan data pada steganografi melalui media citra?
2. Bagaimana cara mengefisiensikan waktu *embedding* dan *extracting* pada steganografi melalui media citra?

### 1.3 Batasan Permasalahan

Batasan dari permasalahan yang dibahas pada Tugas Akhir ini adalah sebagai berikut:

1. Masukan dari implementasi Tugas Akhir ini adalah *grayscale image* berukuran 8-bit dengan dimensi 512x512 *pixel*.
2. Keluaran dari implementasi Tugas Akhir ini adalah *stego image*.
3. Metode yang digunakan pada Tugas Akhir ini adalah metode *Controlling Expansion* [2].
4. Data uji yang diujikan berupa *cover image* dan daftar *payload*.

### 1.4 Tujuan

Tujuan dari pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Mengimplementasikan penyembunyian data melalui *file* citra.
2. Mendapatkan metode untuk meningkatkan kapasitas penyisipan data pada *stego image* dengan memodifikasi metode *Controlling Expansion* [2].

### 1.5 Manfaat

Manfaat dari hasil pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Memberikan sumbangsih dalam bidang steganografi terutama untuk metode penyembunyian data melalui media citra.
2. Menerapkan ilmu yang didapat dari Departemen Informatika ITS supaya dapat berguna bagi masyarakat.

### 1.6 Metodologi

Tahap-tahap yang dilakukan dalam pengerjaan Tugas Akhir ini adalah sebagai berikut:

### **1.6.1 Penyusunan Proposal Tugas Akhir**

Proposal Tugas Akhir disusun sebagai awalan dalam tahapan pengerjaan Tugas Akhir. Pada proposal Tugas Akhir telah diajukan suatu metode pengembangan dari *Controlling Expansion* [2] untuk meningkatkan kapasitas penyisipan data dengan mendapatkan kualitas *stego image* yang optimal.

### **1.6.2 Studi Literatur**

Studi literatur dilakukan untuk mendapatkan referensi pendukung pengerjaan Tugas Akhir. Referensi yang dicari adalah metode steganografi yang digunakan, metode pengolahan *grayscale image* berukuran 8-bit, metode pengolahan berkas uji coba, serta metode statistika seperti *Mean Squared Error (MSE)*, *Peak Signal-to-Noise Ratio (PSNR)*, dan *correlation*.

### **1.6.3 Implementasi Sistem**

Implementasi merupakan tahapan pengimplementasian metode-metode yang digunakan dalam Tugas Akhir. Implementasi perangkat lunak dari Tugas Akhir ini dibangun dengan bahasa pemrograman Python dengan distribusi package Anaconda dan pustaka (*library*) OpenCV. PyCharm sebagai *Integrated Development Environment (IDE)* untuk bahasa pemrograman Python juga digunakan pada pengerjaan perangkat lunak.

### **1.6.4 Pengujian dan Evaluasi**

Pengujian dilakukan dengan mengimplementasikan Tugas Akhir pada perangkat lunak untuk mengevaluasi metode yang diterapkan dalam Tugas Akhir dengan data uji coba yang telah

ditentukan. Hasil yang dievaluasi berupa ukuran data yang disisipkan serta kemiripan *stego image* dengan *cover image*.

### 1.6.5 Penyusunan Laporan Tugas Akhir

Laporan Tugas Akhir disusun dalam bentuk buku yang berisikan dokumentasi pengerjaan Tugas Akhir dan penyajian hasil yang diperoleh dari pengerjaan Tugas Akhir.

## 1.7 Sistematika Penulisan Laporan

Sistematika penulisan laporan Tugas Akhir adalah sebagai berikut:

### 1. Bab I. Pendahuluan

Bab ini berisi penjelasan mengenai latar belakang, rumusan masalah, batasan permasalahan, tujuan, manfaat, metodologi, serta sistematika penulisan laporan Tugas Akhir.

### 2. Bab II. Tinjauan Pustaka

Bab ini berisi teori dan referensi yang digunakan dalam Tugas Akhir. Secara umum, bab ini berisi tentang steganografi, *Difference Expansion (DE)*, *Reduced Difference Expansion (RDE)*, *Controlling Expansion*, Python, Anaconda, OpenCV, *Mean Squared Error (MSE)*, *Peak Signal-to-Noise Ratio (PSNR)*, *correlation*, scikit-image, NumPy, dan Tkinter.

### 3. Bab III. Perancangan Perangkat Lunak

Bab ini menjelaskan perancangan dari modifikasi algoritma *Controlling Expansion* yang diterapkan untuk steganografi pada citra menggunakan bahasa pemrograman Python.

### 4. Bab IV. Implementasi

Bab ini menjelaskan implementasi dan *pseudocode* dari modifikasi algoritma *Controlling Expansion* yang diterapkan untuk steganografi pada citra menggunakan bahasa pemrograman Python.

### 5. Bab V. Hasil Uji Coba dan Evaluasi



Bab ini menyajikan hasil uji coba dari implementasi modifikasi algoritma *Controlling Expansion* yang diterapkan untuk steganografi pada citra menggunakan bahasa pemrograman Python. Pengujian dilakukan menggunakan berkas *grayscale image* berukuran 8-bit. Hasil evaluasi mencakup kapasitas data yang dapat disisipkan serta kemiripan *stego image* dengan *cover image*.

6. Bab VI. Kesimpulan dan Saran

Bab ini memuat kesimpulan dari Tugas Akhir yang dikerjakan serta saran untuk pengembangan metode ke depannya.

7. Daftar Pustaka

Daftar pustaka memuat daftar literatur yang digunakan dalam Tugas Akhir.

8. Lampiran

Lampiran memuat kumpulan *cover image* dan sebagian *stego image*.

*[Halaman ini sengaja dikosongkan]*

## BAB II TINJAUAN PUSTAKA

Bab ini berisi penjelasan mengenai dasar-dasar teori dan referensi yang digunakan dalam Tugas Akhir. Teori-teori tersebut diantaranya adalah algoritma steganografi *Difference Expansion (DE)*, *Reduced Difference Expansion (RDE)*, *Controlling Expansion*, dan beberapa referensi lain yang mendukung pembuatan Tugas Akhir. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap referensi dari riset yang dikembangkan serta pengimplementasiannya dalam bentuk perangkat lunak.

### 2.1 Steganografi

Steganografi berasal dari Bahasa Yunani yang berarti *covered writing*. Dengan kata lain, steganografi merupakan suatu seni untuk melakukan komunikasi secara tersembunyi [3]. Tujuan dari steganografi adalah untuk menyembunyikan data pada media tertentu dan orang yang tidak memiliki hak akses pada data tersebut tidak dapat mengetahui bahwa terdapat data yang disisipkan pada media steganografi [1].

### 2.2 *Difference Expansion*

*Difference Expansion (DE)* merupakan algoritma *reversible steganography* yang diusulkan oleh Tian [4]. Tahap-tahap dari algoritma *DE* adalah:

1. Menghitung *average (l)* dan *difference (h)* dari sepasang *pixel x* dan *y* menggunakan Persamaan (2.1).

$$l = \left\lfloor \frac{x + y}{2} \right\rfloor, h = x - y \quad (2.1)$$

2. Nilai *pixel* tidak boleh *overflow* atau *underflow* dengan harus memenuhi Persamaan (2.2) dan Persamaan (2.3).

$$|h| \leq 2[255 - l] \text{ and } |h| \leq 2l + 1 \quad (2.2)$$

$$\begin{cases} |h| \leq 2(255 - l), \text{ if } 128 \leq x \leq 255 \\ |h| \leq 2l + 1, \text{ if } 0 \leq x \leq 127 \end{cases} \quad (2.3)$$

3. Setelah dipastikan nilai *pixel* tidak *overflow* atau *underflow*, maka dilanjutkan dengan menyisipkan *payload* (*b*) dalam bentuk *binary* menggunakan Persamaan (2.4).

$$h' = 2h + b \quad (2.4)$$

4. Mendapatkan sepasang *stego pixel*  $x'$  dan  $y'$  melalui Persamaan (2.5).

$$x' = l + \left\lfloor \frac{h' + 1}{2} \right\rfloor, y' = l - \left\lfloor \frac{h'}{2} \right\rfloor \quad (2.5)$$

5. Untuk mengembalikan nilai *b* dan *h*, gunakan Persamaan (2.6).

$$b = LSB(h'), h = \left\lfloor \frac{h'}{2} \right\rfloor \quad (2.6)$$

6. Untuk mengembalikan nilai sepasang *original pixel* *x* dan *y*, gunakan Persamaan (2.7).

$$x = l + \left\lfloor \frac{h + 1}{2} \right\rfloor, y = l - \left\lfloor \frac{h}{2} \right\rfloor \quad (2.7)$$

### 2.3 Reduced Difference Expansion

*Reduced Difference Expansion (RDE)* merupakan algoritma *reversible steganography* sebagai pengembangan dari algoritma

*DE* [4]. Algoritma *RDE* diusulkan oleh Lou et al. [5]. Tahap-tahap algoritma *RDE* melakukan penyisipan *payload* adalah:

1. Menghitung *average* ( $l$ ) dan *difference* ( $h$ ) dari sepasang *pixel*  $x$  dan  $y$  menggunakan Persamaan (2.1).
2. Melakukan reduksi  $h$  menggunakan Persamaan (2.8).

$$h' = \begin{cases} h, & \text{if } h' < 2 \\ h - 2^{\lfloor \log_2 h \rfloor - 1}, & \text{else} \end{cases} \quad (2.8)$$

3. Menentukan location map ( $LM$ ) menggunakan Persamaan (2.9).

$$LM = \begin{cases} 0, & \text{if } 2^{\lfloor \log_2 h' \rfloor - 1} = 2^{\lfloor \log_2 h \rfloor} \text{ or } h' = h \\ 1, & \text{if } 2^{\lfloor \log_2 h' \rfloor} \neq 2^{\lfloor \log_2 h \rfloor} \end{cases} \quad (2.9)$$

4. Menyisipkan *payload* ( $b$ ) dalam bentuk *binary* menggunakan Persamaan (2.10).

$$h'' = 2h + b \quad (2.10)$$

5. Didapatkan sepasang *stego pixel*  $x'$  dan  $y'$  menggunakan Persamaan (2.11).

$$x' = l + \left\lfloor \frac{h'' + 1}{2} \right\rfloor, y' = l - \left\lfloor \frac{h''}{2} \right\rfloor \quad (2.11)$$

Sedangkan langkah untuk melakukan ekstraksi *payload* dengan algoritma *RDE* adalah:

1. Menghitung *average* ( $l$ ) dan *difference* ( $h''$ ) dari pasangan *pixel*  $x'$  dan  $y'$  menggunakan Persamaan (2.12).

$$l = \left\lfloor \frac{x' + y'}{2} \right\rfloor, h'' = x' - y' \quad (2.12)$$

2. Mengembalikan nilai *payload* ( $b$ ) menggunakan Persamaan (2.13).

$$b = LSB(h'') \quad (2.13)$$

3. Menghitung nilai  $h'$  menggunakan Persamaan (2.14).

$$h' = \left\lfloor \frac{h''}{2} \right\rfloor \quad (2.14)$$

4. Mengembalikan nilai *difference* ( $h$ ) dari *original pixel*  $x$  dan  $y$  menggunakan Persamaan (2.15).

$$h = \begin{cases} h' + 2^{\lfloor \log_2 h' \rfloor - 1}, & \text{if } LM = 0 \\ h' + 2^{\lfloor \log_2 h' \rfloor}, & \text{if } LM = 1 \end{cases} \quad (2.15)$$

5. Mengembalikan pasangan *original pixel*  $x$  dan  $y$  menggunakan Persamaan (2.16).

$$x = l + \left\lfloor \frac{h+1}{2} \right\rfloor, y = l - \left\lfloor \frac{h}{2} \right\rfloor \quad (2.16)$$

## 2.4 Controlling Expansion

*Controlling Expansion* adalah algoritma *reversible steganography* yang diusulkan oleh Angreni dan Ahmad [2] dan merupakan pengembangan dari algoritma *DE* [4] dan *RDE* [5]. Tahap-tahap dalam algoritma *Controlling Expansion* adalah:

1. Menghitung *difference* ( $h$ ) dan *average* ( $l$ ) dari *pixel*  $x$  dan nilai *random*  $R$  menggunakan Persamaan (2.17).

$$h = x - R, l = \begin{cases} \left\lfloor \frac{x+R}{2} \right\rfloor, & \text{if } h \leq 1 \\ \left\lceil \frac{x+R}{2} \right\rceil, & \text{if } h > 1 \end{cases} \quad (2.17)$$

2. Melakukan reduksi  $h$  menggunakan Persamaan (2.18).

$$h' = \begin{cases} |h|, & \text{if } h = 0 \text{ or } h = 1 \\ \left\lfloor \frac{|h| - 1}{2} \right\rfloor, & \text{else} \end{cases} \quad (2.18)$$

3. Menyisipkan *payload* ( $b$ ) dalam bentuk *binary* menggunakan Persamaan (2.19).

$$h'' = \begin{cases} 2h' - b, & \text{if } h < 0 \\ 2h' + b, & \text{if } h \geq 0 \end{cases} \quad (2.19)$$

4. Mendapatkan *stego pixel*  $x'$  melalui Persamaan (2.20).

$$x' = \begin{cases} l - \left\lfloor \frac{h'' + 1}{2} \right\rfloor, & \text{if } h < 0 \text{ and } x = 2n + 1 \\ l - \left\lfloor \frac{h''}{2} \right\rfloor, & \text{if } h < 0 \text{ and } x = 2n \\ l + \left\lfloor \frac{h'' - 1}{2} \right\rfloor, & \text{if } h \geq 0 \text{ and } x = 2n + 1 \\ l + \left\lfloor \frac{h''}{2} \right\rfloor, & \text{if } h \geq 0 \text{ and } x = 2n \end{cases} \quad (2.20)$$

5. Mendapatkan *location map* ( $LM$ ) melalui Persamaan (2.21).

$$LM = \begin{cases} 1, & \text{if } x = 2n + 1 \\ 0, & \text{if } x = 2n \end{cases} \quad (2.21)$$

6. Untuk mengembalikan nilai *payload* ( $b$ ), gunakan Persamaan (2.22).

$$b = LSB(x') \quad (2.22)$$

7. Untuk mengembalikan nilai *original pixel* ( $x$ ), ganti nilai  $b$  dengan  $LM$  dan ganti nilai  $x$  dengan  $x'$ , lalu ulangi langkah dari 1 sampai 4 sehingga didapatkan nilai *original pixel* hasil dari Persamaan (2.20).

## 2.5 Python

Python adalah suatu bahasa pemrograman *interpreter* yang telah mendukung pemrograman berorientasi obyek dan termasuk dalam bahasa pemrograman tingkat tinggi [6]. Saat ini Python dapat digunakan di banyak sistem operasi, seperti pada banyak varian Unix, macOS, dan Windows [7]. Bahasa pemrograman Python bersifat *open source* [8]. Logo Python ditunjukkan pada Gambar 2.1 Logo Python.



**Gambar 2.1 Logo Python [9]**

## 2.6 Anaconda

Anaconda adalah suatu *platform* untuk *data science* pada bahasa pemrograman Python dan R. Anaconda menyediakan fitur untuk manajemen *environment* bahasa pemrograman Python dan R. Saat ini Anaconda dapat digunakan pada sistem operasi Windows, macOS, dan Linux [10]. Logo Anaconda ditunjukkan pada Gambar 2.2 Logo Anaconda.





**Gambar 2.2 Logo Anaconda [11]**

## **2.7 OpenCV**

*Open Source Computer Vision (OpenCV)* adalah pustaka yang bersifat *open source* untuk visi computer dan *machine learning* yang ditulis dalam Bahasa C++. OpenCV menyediakan lebih dari 2500 algoritma visi komputer dan *machine learning*. OpenCV dapat digunakan di bahasa pemrograman C++, C, Python, Java, dan MATLAB serta pada sistem operasi Windows, Linux, Android, dan macOS [12]. Logo OpenCV ditunjukkan pada Gambar 2.3 Logo OpenCV.



**Gambar 2.3 Logo OpenCV [13]**

## **2.8 Mean Squared Error**

*Mean Squared Error (MSE)* adalah perbedaan kuadrat rata-rata antara nilai *pixel* pada *stego image* dan nilai *pixel* pada *cover image*. *MSE* dari *cover image*  $X$  dan *stego image*  $X'$  dengan dimensi  $m \times n$  dapat dihitung menggunakan Persamaan (2.23).

$$MSE = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n (X_{ij} - X'_{ij})^2 \quad (2.23)$$

## 2.9 Peak Signal-to-Noise Ratio

*Peak Signal-to-Noise Ratio (PSNR)* adalah perbandingan antara nilai maksimum dari kedalaman bit citra yang diukur dibandingkan dengan derau yang diwakili oleh *MSE* [14]. *PSNR* digunakan untuk mengukur tingkat persamaan antara *stego image* dan *cover image* [15] yang biasanya dinyatakan dalam skala desibel (dB). Semakin besar nilai *PSNR*, maka semakin besar tingkat persamaan antara *stego image* dan *cover image*. *PSNR* dapat dihitung menggunakan Persamaan (2.24).

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \quad (2.24)$$

## 2.10 Correlation

*Correlation* digunakan untuk mengukur tingkat kemiripan antara dua data. Nilai *correlation* berkisar antara -1 sampai 1 yang mana jika nilai *correlation* semakin mendekati 1, maka data yang dibandingkan akan semakin mirip [16]. *Correlation* dapat dihitung menggunakan Persamaan (2.25) dengan  $X$  adalah data pertama dan  $Y$  adalah data kedua serta  $\bar{X}$  adalah rata-rata dari data pertama dan  $\bar{Y}$  adalah rata-rata dari data kedua.

$$\begin{aligned} & \text{Correlation} \\ &= \frac{\sum_{i=1}^n (X_i - \bar{X}) \times (Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \times \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}} \end{aligned} \quad (2.25)$$

## 2.11 scikit-image

scikit-image adalah pustaka yang menyediakan algoritma pengolahan citra untuk bahasa pemrograman Python [17]. Logo scikit-image ditunjukkan pada Gambar 2.4 Logo scikit-image.



**Gambar 2.4 Logo scikit-image [18]**

## 2.12 NumPy

NumPy adalah pustaka yang menyediakan fitur untuk *scientific computing* pada bahasa pemrograman Python. Selain itu, Numpy juga dapat digunakan sebagai *multidimensional container* yang efisien pada *generic data* [19]. Logo Numpy ditunjukkan pada Gambar 2.5 Logo Numpy.



**Gambar 2.5 Logo Numpy [20]**

## 2.13 Tkinter

Tkinter merupakan *standard library* pada bahasa pemrograman Python yang digunakan untuk membuat *Graphical User Interface (GUI)*. Tkinter dapat berjalan pada sebagian besar sistem operasi yang berbasis Unix dan sistem operasi Windows [21]. Logo Tkinter ditunjukkan pada Gambar 2.6 Logo Tkinter.



**Gambar 2.6 Logo Tkinter [22]**

## **BAB III**

### **PERANCANGAN PERANGKAT LUNAK**

Bab ini membahas mengenai perancangan modifikasi dari algoritma *Controlling Expansion* [2] yang dibuat pada Tugas Akhir.

#### **3.1 Data**

Pada sub bab ini akan dijelaskan data yang dipakai dalam implementasi metode. Data tersebut akan digunakan dalam proses pengujian dan evaluasi metode.

##### **3.1.1 Data Masukan**

Data masukan adalah data dalam bentuk *cover image* dan *payload* yang digunakan dalam proses penyisipan pesan. Data masukan yang digunakan merupakan *grayscale image* berukuran 8-bit yang berperan sebagai *cover image* dan daftar *payload* dalam bentuk *file* berekstensi .txt.

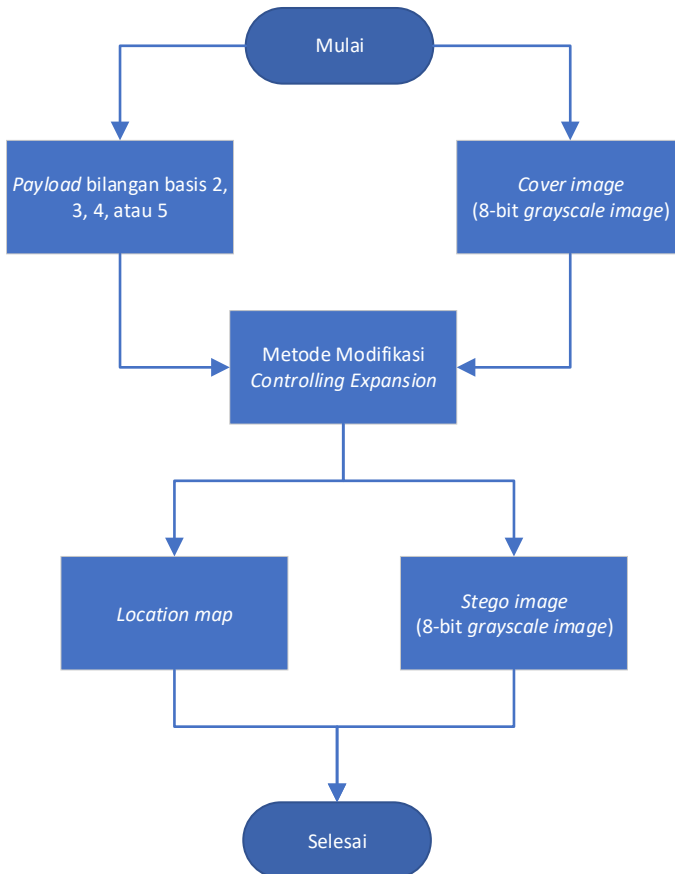
##### **3.1.2 Data Keluaran**

Data keluaran adalah data dalam bentuk *stego image* dan *location map* yang dihasilkan dari proses penyisipan pesan. Data masukan yang dihasilkan merupakan *stego image* yang berupa *grayscale image* berukuran 8-bit dan *location map* berupa *file* berekstensi .txt.

#### **3.2 Deskripsi Umum**

Dalam Tugas Akhir ini penulis akan mengimplementasikan modifikasi dari algoritma *Controlling Expansion* [2] untuk meningkatkan kapasitas *payload* yang dapat disisipkan. Pada metode sebelumnya [2], jenis *payload* pada proses penyisipan

menggunakan bilangan biner (basis 2). Pada Tugas Akhir ini, *payload* dapat dimasukkan sampai dengan bilangan basis 5. Untuk lebih jelasnya dapat dilihat dalam Gambar 3.1 Diagram Alir Gambaran Umum Modifikasi Algoritma *Controlling Expansion*.



**Gambar 3.1 Diagram Alir Gambaran Umum Modifikasi Algoritma *Controlling Expansion***

Peningkatan nilai basis bilangan pada *payload* akan meningkatkan kapasitas penyisipan data. Misalkan pada metode sebelumnya [2] terdapat daftar *payload* dalam bentuk bilangan basis 2 bernilai 101011010111 yang mana panjang *payload* adalah 12 digit. *Payload* tersebut lalu dikonversikan ke dalam bilangan basis 5 sehingga nilai *payload* menjadi 42100 yang terdiri dari 5 digit. Penyusutan panjang *payload* dari 12 digit menjadi 5 digit menunjukkan bahwa peningkatan nilai basis bilangan dari bilangan basis 2 menjadi bilangan basis 5 dapat meningkatkan kapasitas *payload* yang akan disisipkan.

### 3.3 Perancangan Penyisipan *Payload*

Modifikasi dari algoritma *Controlling Expansion* [2] yang dilakukan memungkinkan *payload* disisipkan dalam bentuk bilangan basis 2, 3, 4, dan 5. *Payload* pada data masukan merupakan data dalam bentuk bilangan basis 2. Jika terdapat nilai 0 pada awal *payload* basis 2, maka nilai 0 akan diabaikan. Bilangan basis 2 tersebut nantinya akan dikonversikan ke dalam bilangan bentuk desimal (basis 10), lalu bilangan bentuk desimal akan dikonversikan ke dalam bilangan basis 3, 4, dan 5. Misalkan terdapat *payload* 101011010111 yang akan dikonversikan ke dalam bentuk bilangan basis 5. Maka yang pertama akan dikonversikan ke dalam bentuk desimal sebagai berikut,

$$\begin{aligned}
 (101011010111)_2 &= 1 \times 2^{11} + 0 \times 2^{10} + 1 \times 2^9 + 0 \times 2^8 \\
 &+ 1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 \\
 &+ 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = (2775)_{10}
 \end{aligned}$$

dari perhitungan di atas didapatkan bahwa  $(101011010111)_2 = (2775)_{10}$ .

Bilangan desimal tersebut lalu dikonversikan ke dalam bentuk bilangan basis 3, 4, dan 5. Konversi dari bilangan desimal ke bilangan basis 3 adalah sebagai berikut,

$$\begin{aligned}
2775 \text{ dibagi } 3 &= 925 \text{ sisa } \mathbf{0} \\
925 \text{ dibagi } 3 &= 308 \text{ sisa } \mathbf{1} \\
308 \text{ dibagi } 3 &= 102 \text{ sisa } \mathbf{2} \\
102 \text{ dibagi } 3 &= 34 \text{ sisa } \mathbf{0} \\
34 \text{ dibagi } 3 &= 11 \text{ sisa } \mathbf{1} \\
11 \text{ dibagi } 3 &= 3 \text{ sisa } \mathbf{2} \\
3 \text{ dibagi } 3 &= \mathbf{1} \text{ sisa } \mathbf{0}
\end{aligned}$$

sehingga didapatkan  $(2775)_{10} = (10210210)_3$ .

Berikutnya dilakukan konversi dari bilangan desimal ke bilangan basis 4 sebagai berikut,

$$\begin{aligned}
2775 \text{ dibagi } 4 &= 693 \text{ sisa } \mathbf{3} \\
693 \text{ dibagi } 4 &= 173 \text{ sisa } \mathbf{1} \\
173 \text{ dibagi } 4 &= 43 \text{ sisa } \mathbf{1} \\
43 \text{ dibagi } 4 &= 10 \text{ sisa } \mathbf{3} \\
10 \text{ dibagi } 4 &= \mathbf{2} \text{ sisa } \mathbf{2}
\end{aligned}$$

sehingga didapatkan  $(2775)_{10} = (223113)_4$ .

Selanjutnya dilakukan konversi dari bilangan desimal ke bilangan basis 5 sebagai berikut,

$$\begin{aligned}
2775 \text{ dibagi } 5 &= 555 \text{ sisa } \mathbf{0} \\
555 \text{ dibagi } 5 &= 111 \text{ sisa } \mathbf{0} \\
111 \text{ dibagi } 5 &= 22 \text{ sisa } \mathbf{1} \\
22 \text{ dibagi } 5 &= \mathbf{4} \text{ sisa } \mathbf{2}
\end{aligned}$$

sehingga didapatkan  $(2775)_{10} = (42100)_5$

Dari perhitungan konversi bilangan desimal ke bilangan basis 3, 4, dan 5, maka didapatkan  $(101011010111)_2 = (10210210)_3 = (223113)_4 = (42100)_5$ .

### 3.4 Modifikasi Algoritma *Controlling Expansion*



Pada Tugas Akhir ini akan dilakukan modifikasi dari algoritma steganografi *Controlling Expansion* [2]. Modifikasi dilakukan untuk meningkatkan kapasitas penyisipan data. Pada metode *Controlling Expansion*, data disisipkan dalam bentuk bilangan basis 2. Dalam Tugas Akhir ini akan dilakukan peningkatan basis bilangan untuk meningkatkan kapasitas penyisipan data. Setelah didapatkan metode peningkatan penyisipan data, akan dilakukan evaluasi pada kapasitas data yang disisipkan serta kemiripan stego image dengan citra asli. Langkah-langkah modifikasi algoritma yang dilakukan adalah:

1. Menghitung *difference* ( $h$ ) dan *average* ( $l$ ) dari *pixel*  $x$  dan nilai *random*  $R$  menggunakan Persamaan (2.17) yang diambil dari *Controlling Expansion* [2].
2. Melakukan reduksi  $h$  menggunakan Persamaan (2.18) yang diambil dari *Controlling Expansion* [2] sehingga didapatkan  $h'$ .
3. *Payload* ( $b$ ) disisipkan dalam bentuk bilangan basis 2, 3, 4, atau 5 dengan cara menyisipkan *payload* dalam bentuk *binary* untuk nilai *payload* 0 dan 1, serta menyisipkan *payload* dalam bentuk *ternary* untuk nilai *payload* 2, 3, dan 4 dengan mengganti nilai *payload* 2 menjadi 1, 3 menjadi 2, dan 4 menjadi 3 seperti pada Persamaan (3.1) sehingga didapatkan nilai  $b'$ . Penyisipan *payload* dilakukan menggunakan Persamaan (3.2) sehingga didapatkan nilai  $h''$ .

$$b' = \begin{cases} 0, & \text{if } b = 0 \text{ or } b = 2 \\ 1, & \text{if } b = 1 \text{ or } b = 3 \\ 2, & \text{if } b = 4 \end{cases} \quad (3.1)$$

$$h'' = \begin{cases} 2h' - b', & \text{if } h < 0 \\ 2h' + b', & \text{if } h \geq 0 \end{cases} \quad (3.2)$$

4. Mendapatkan *stego pixel* ( $x'$ ) menggunakan Persamaan (3.3) dan Persamaan (3.4).

$$a = \begin{cases} 1, & \text{if } b = 3 \text{ or } b = 4 \\ 2, & \text{if } b = 2 \end{cases}, \text{ if } x \bmod 3 = 2 \quad (3.3)$$

$$\begin{cases} 0, & \text{if } b = 2 \text{ or } b = 3 \\ 1, & \text{if } b = 4 \end{cases}, \text{ if } x \bmod 3 < 2$$

$$x' = \begin{cases} l - \left\lceil \frac{h'' + 1}{2} \right\rceil, & \text{if } h < 0 \text{ and } x \bmod 2 = 1 \\ l - \left\lceil \frac{h''}{2} \right\rceil, & \text{if } h < 0 \text{ and } x \bmod 2 = 0 \\ l + \left\lceil \frac{h'' - 1}{2} \right\rceil, & \text{if } h \geq 0 \text{ and } x \bmod 2 = 1 \\ l + \left\lceil \frac{h''}{2} \right\rceil, & \text{if } h \geq 0 \text{ and } x \bmod 2 = 0 \end{cases}, \text{ if } b \leq 1 \quad (3.4)$$

$$= \begin{cases} l - \left\lceil \frac{h''}{2} \right\rceil - a, & \text{if } x < 255 \text{ and } h < 0 \text{ and } x \bmod 3 = 2 \\ l - \left\lceil \frac{h'' + 1}{2} \right\rceil + a, & \text{if } x < 255 \text{ and } h < 0 \text{ and } x \bmod 3 = 1 \\ l - \left\lceil \frac{h''}{2} \right\rceil + a, & \text{if } x < 255 \text{ and } h < 0 \text{ and } x \bmod 3 = 0 \\ l + \left\lceil \frac{h''}{2} \right\rceil - a, & \text{if } x < 255 \text{ and } h \geq 0 \text{ and } x \bmod 3 = 2 \\ l + \left\lceil \frac{h'' - 1}{2} \right\rceil + a, & \text{if } x < 255 \text{ and } h \geq 0 \text{ and } x \bmod 3 = 1 \\ l + \left\lceil \frac{h''}{2} \right\rceil + a, & \text{if } x < 255 \text{ and } h \geq 0 \text{ and } x \bmod 3 = 0 \\ 255, & \text{if } x = 255 \end{cases}, \text{ if } b > 1$$

5. Untuk mendapatkan *location map (LM)*, gunakan Persamaan (3.5).

$$LM = \begin{cases} 00, & \text{if } x \bmod 2 = 0 \\ 01, & \text{if } x \bmod 2 = 1 \end{cases}, \text{ if } b \leq 1 \quad (3.5)$$

$$\begin{cases} 12, & \text{if } (x < 255 \text{ and } x \bmod 3 = 2) \text{ or } (x = 255 \text{ and } b = 4) \\ 11, & \text{if } (x < 255 \text{ and } x \bmod 3 = 1) \text{ or } (x = 255 \text{ and } b = 3), \text{ if } b > 1 \\ 10, & \text{if } (x < 255 \text{ and } x \bmod 3 = 0) \text{ or } (x = 255 \text{ and } b = 2) \end{cases}$$

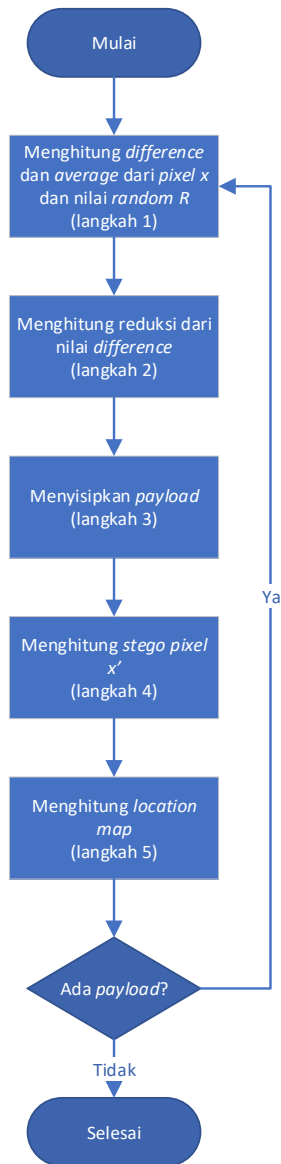
6. Untuk mengembalikan nilai *payload (b)*, gunakan Persamaan (3.6).

$$b = \begin{cases} LSB(x'), \text{ if } LM[0] = 0 \\ 2, \text{ if } (x' < 255 \text{ and } x' \bmod 3 = 0) \text{ or } (x' = 255 \text{ and } LM[1] = 0) \\ 3, \text{ if } (x' < 255 \text{ and } x' \bmod 3 = 1) \text{ or } (x' = 255 \text{ and } LM[1] = 1), \text{ if } LM[0] = 1 \\ 4, \text{ if } (x' < 255 \text{ and } x' \bmod 3 = 2) \text{ or } (x' = 255 \text{ and } LM[1] = 2) \end{cases} \quad (3.6)$$

7. Untuk mengembalikan nilai *original pixel* ( $x$ ), ganti nilai  $b$  dengan hasil dari Persamaan (3.7) dan ganti nilai  $x$  dengan  $x'$ . Setelah itu ulangi dari langkah 1 sampai 4 sehingga didapatkan nilai *original pixel* hasil dari Persamaan (3.4).

$$b = \begin{cases} LM[1], \text{ if } LM[0] = 0 \\ 2, \text{ if } LM[0] = 1 \text{ and } LM[1] = 0 \\ 3, \text{ if } LM[0] = 1 \text{ and } LM[1] = 1 \\ 4, \text{ if } LM[0] = 1 \text{ and } LM[1] = 2 \end{cases} \quad (3.7)$$

Diagram alir dari modifikasi algoritma *Controlling Expansion* terdapat pada Gambar 3.2 Diagram Alir Penyisipan *Payload* untuk proses penyisipan *payload* dan Gambar 3.3 Diagram Alir Ekstraksi *Payload* untuk proses pengembalian *payload*.



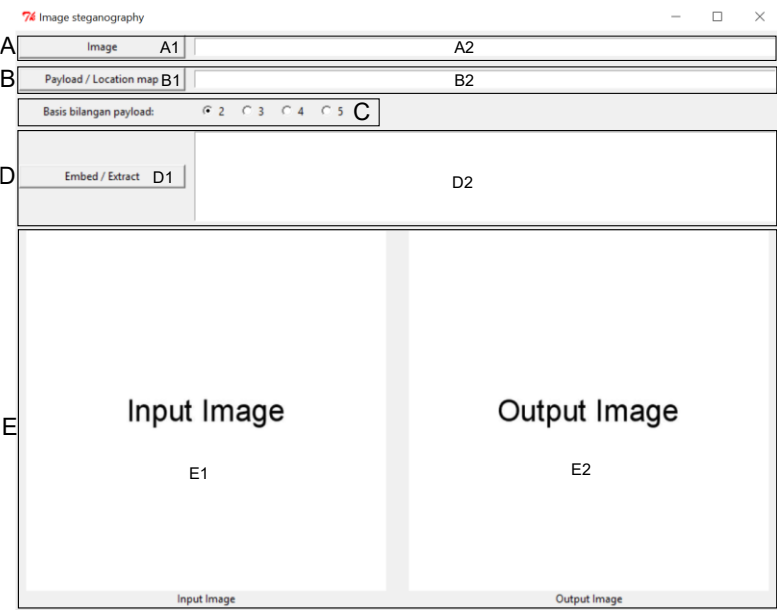
**Gambar 3.2 Diagram Alir Penyisipan *Payload***



**Gambar 3.3 Diagram Alir Ekstraksi Payload**

3.5 Perancangan Antarmuka

Pembuatan antarmuka bertujuan untuk memudahkan pengguna dalam menerapkan metode steganografi yang digunakan dalam Tugas Akhir ini. Antarmuka dapat digunakan untuk mengimplementasikan penyisipan dan ekstraksi *payload*. Antarmuka dibuat menggunakan *library* Tkinter yang merupakan *standard library* untuk pembuatan antarmuka pada bahasa pemrograman Python [21]. Tampilan antarmuka yang dibuat dalam Tugas Akhir ini dapat dilihat pada Gambar 3.4 Desain Antarmuka.



Gambar 3.4 Desain Antarmuka

Komponen yang terdapat dalam Gambar 3.4 Desain Antarmuka antara lain:

- A. Bagian A berisikan menu untuk memilih citra yang akan digunakan dalam proses steganografi. A1 merupakan tombol yang akan menampilkan *file dialog* untuk memilih berkas citra dan A2 akan menampilkan alamat dari berkas citra yang telah dipilih.
- B. Bagian B berisikan menu untuk memilih *payload* yang akan digunakan dalam proses steganografi. B1 merupakan tombol yang akan menampilkan *file dialog* untuk memilih berkas *payload* dan A2 akan menampilkan alamat dari berkas *payload* yang telah dipilih.
- C. Bagian C berisikan pilihan untuk basis bilangan *payload*. Pilihan terdiri dari bilangan basis 2, 3, 4, dan 5 yang disajikan dalam bentuk *radio button*.
- D. Bagian D berisikan menu untuk melakukan proses steganografi. D1 merupakan tombol untuk memulai proses steganografi dan D2 akan menampilkan informasi dari hasil proses steganografi.
- E. Bagian E berisikan tampilan untuk citra yang digunakan dalam proses steganografi. E1 akan menampilkan citra masukan dan E2 akan menampilkan citra keluaran.

*[Halaman ini sengaja dikosongkan]*



## BAB IV IMPLEMENTASI

Bab ini membahas implementasi dari perancangan metode yang telah dijelaskan pada bab sebelumnya. Kode program diimplementasikan menggunakan bahasa pemrograman Python.

### 4.1 Lingkungan Implementasi

Lingkungan implementasi penyisipan dan ekstraksi *payload* pada media citra ditunjukkan pada Tabel 4.1 Lingkungan Implementasi Perangkat Lunak.

**Tabel 4.1 Lingkungan Implementasi Perangkat Lunak**

Perangkat	Jenis Perangkat	Spesifikasi
Perangkat Keras	Prosesor	Intel® Core™ i5-8250U CPU @ 1.80 GHz x 4
	Memori	8 GB 2400 MHz DDR4
	Penyimpanan	1 TB Hard Drive
Perangkat Lunak	Sistem Operasi	Windows 10 Home 64-bit
	Perangkat Pengembang	PyCharm Community 2018.1.3

### 4.2 Implementasi

#### 4.2.1 Implementasi Pembacaan Citra

Pembacaan citra menghasilkan data *pixel* beserta dimensi citra dengan ukuran  $rows \times cols$ . Pembacaan citra diimplementasikan untuk membaca berkas citra *grayscale img\_file* berukuran 8-bit menggunakan pustaka OpenCV. *Pseudocode* untuk pembacaan citra *grayscale* berukuran 8-bit menggunakan pustaka OpenCV dapat dilihat pada *Pseudocode 4.1* Implementasi Pembacaan Berkas Citra *Grayscale 8-bit* dengan Pustaka OpenCV.

```
1. img <- cv2.imread(img_file, cv2.IMREAD_GRAYSCALE)
2. rows, cols <- img.shape[:2]
```

***Pseudocode 4.1 Impelementasi Pembacaan Berkas Citra  
Grayscale 8-bit dengan Pustaka OpenCV***

#### **4.2.2 Implementasi Pengukuran Waktu Penyisipan/Ekstraksi *Payload***

Pengukuran waktu diimplementasikan menggunakan fungsi *time*. *Pseudocode* dari pengukuran waktu penyisipan/ekstraksi *payload* ditunjukkan pada *Pseudocode 4.2 Implementasi Pengukuran Waktu Penyisipan/Ekstraksi *Payload**.

```
1. start_time <- time.time()
2. end_time <- time.time() - start_time
```

***Pseudocode 4.2 Implementasi Pengukuran Waktu  
Penyisipan/Ekstraksi *Payload****

#### **4.2.3 Implementasi Penentuan Nilai *Random***

Nilai *random* diimplementasikan menggunakan fungsi *randint* dengan rentang nilai antara -999999 sampai dengan 999999. *Pseudocode* dari penentuan nilai *random* ditunjukkan pada *Pseudocode 4.3 Implementasi Penentuan Nilai *Random**.

```
1. random_num <- random.randint(-999999, 999999)
```

***Pseudocode 4.3 Implementasi Penentuan Nilai *Random****

#### **4.2.4 Implementasi Perhitungan *Correlation***

Perhitungan *correlation* dilakukan untuk *payload* dan citra. Perhitungan *correlation* menggunakan fungsi *corref* yang diambil dari pustaka NumPy [19]. *Pseudocode* dari perhitungan *correlation* ditunjukkan pada *Pseudocode 4.4 Impelementasi Perhitungan *Correlation**.

```
1. corr = numpy.corref(inp_file, out_file)
```

#### ***Pseudocode 4.4 Impelementasi Perhitungan Correlation***

##### **4.2.5 Modul *ModifiedControllingExpansion***

Modul *ModifiedControllingExpansion* berbentuk kelas. Kelas *ModifiedControllingExpansion* berisikan fungsi utama yang ada pada metode hasil modifikasi algoritma *Controlling Expansion* [2]. Daftar fungsi dalam modul *ModifiedControllingExpansion* dapat dilihat pada Tabel 4.2 Daftar Fungsi Modul *ModifiedControllingExpansion*. Penjelasan detail fungsi yang ada dalam modul *ModifiedControllingExpansion* akan dijelaskan lebih lanjut pada sub poin berikutnya.

**Tabel 4.2 Daftar Fungsi Modul *ModifiedControllingExpansion***

<b>Nama Fungsi</b>	<b>Keterangan</b>
<i>_init_</i>	Menginisiasi data yang dibutuhkan untuk proses penyisipan maupun ekstraksi.
<i>get_difference</i>	Mencari nilai <i>difference</i> dari nilai <i>pixel</i> dan nilai <i>random</i> .
<i>get_average</i>	Mencari nilai <i>average</i> dari nilai <i>difference</i> , nilai <i>pixel</i> , dan nilai <i>random</i> .
<i>get_reduction</i>	Mencari nilai <i>difference</i> dari nilai <i>average</i> .
<i>get_embed</i>	Mencari nilai <i>embed</i> dari nilai <i>difference</i> dan nilai <i>reduction</i> .
<i>get_pixel</i>	Mencari nilai <i>pixel</i> baru dari nilai <i>difference</i> , nilai <i>reduction</i> , dan nilai <i>pixel</i> hasil <i>mapping</i> dari fungsi <i>_init_</i> .
<i>run</i>	Menjalankan seluruh fungsi pada kelas setelah fungsi <i>_init_</i> dijalankan.

#### 4.2.5.1 Implementasi Fungsi *\_init\_*

Fungsi *\_init\_* berfungsi untuk menginisiasi variabel nilai *random*  $r$ , nilai *payload*  $b$ , nilai *pixel*  $x$ , dan nilai *difference*  $h$  yang akan digunakan pada *method* lain pada kelas serta melakukan mapping untuk mendapatkan nilai variabel  $b'$  yang didapat dari nilai  $b$ . *Pseudocode* dari fungsi *\_init\_* ditunjukkan pada *Pseudocode* 4.5 Fungsi *\_init\_*.

```

1. FUNCTION __init__(self, r, b, x):
2.     self.b <- b
3.     self.x <- x
4.     self.r <- r
5.     self.h <- self.get_difference()
6.     IF self.b = 0 OR self.b = 2:
7.         self.b2 <- 0
8.     ELSE IF self.b = 1 OR self.b = 3:
9.         self.b2 <- 1
10.    ELSE IF self.b = 4:
11.        self.b2 <- 2
12.    ELSE IF self.b = 4:
13.        self.b2 <- 0
14.    ENDIF
15. ENDFUNCTION

```

*Pseudocode 4.5 Fungsi \_\_init\_\_*

#### 4.2.5.2 Implementasi Fungsi *get\_difference*

Fungsi *get\_difference* digunakan untuk mencari nilai *difference* (*h*) dari nilai *pixel x* dan nilai *random r*. *Pseudocode* dari fungsi *get\_difference* ditunjukkan pada *Pseudocode 4.6* Fungsi *get\_difference*.

```

1. FUNCTION get_difference(self):
2.     RETURN self.x - self.r
3. ENDFUNCTION

```

*Pseudocode 4.6 Fungsi get\_difference*

#### 4.2.5.3 Implementasi Fungsi *get\_average*

Fungsi *get\_difference* digunakan untuk mencari nilai *average* (*l*) dari nilai *difference h*, nilai *pixel x*, dan nilai *random r*. *Pseudocode* dari fungsi *get\_average* ditunjukkan pada *Pseudocode 4.7* Fungsi *get\_average*.

```

1. FUNCTION get_average(self):
2.     IF self.h <= 1:
3.         RETURN FLOOR((self.x + self.r) / 2)
4.     ELSE IF self.h > 1:
5.         RETURN CEIL((self.x + self.r) / 2)
6.     ENDIF
7. ENDFUNCTION

```

*Pseudocode 4.7 Fungsi get\_average*

#### 4.2.5.4 Implementasi Fungsi *get\_reduction*

Fungsi *get\_reduction* digunakan untuk mencari nilai *reduction* ( $h'$ ) dari nilai *difference*  $h$ . *Pseudocode* dari fungsi *get\_reduction* ditunjukkan pada *Pseudocode 4.8* Fungsi *get\_reduction*.

```

1. FUNCTION get_reduction(self):
2.     IF self.h = 0 OR self.h = 1:
3.         RETURN ABSOLUTE(self.h)
4.     ELSE:
5.         RETURN CEIL((abs(self.h) - 1) / 2)
6.     ENDIF
7. ENDFUNCTION

```

*Pseudocode 4.8 Fungsi get\_reduction*

#### 4.2.5.5 Implementasi Fungsi *get\_embed*

Fungsi *get\_embed* digunakan untuk mencari nilai *embed* ( $h''$ ) dari nilai *difference*  $h$  nilai *reduction*  $h'$ , dan nilai *pixel* mapping  $b'$ . *Pseudocode* dari fungsi *get\_embed* ditunjukkan pada *Pseudocode 4.9* Fungsi *get\_embed*.

```

1. FUNCTION get_embed(self, h2):
2.     IF self.h < 0:
3.         RETURN 2 * h2 - self.b2
4.     ELSE IF self.h >= 0:
5.         RETURN 2 * h2 + self.b2
6.     ENDIF
7. ENDFUNCTION

```

***Pseudocode 4.9 Fungsi get\_embed***

#### **4.2.5.6 Implementasi Fungsi *get\_pixel***

Fungsi *get\_pixel* digunakan untuk mencari nilai *pixel* baru ( $x'$ ) dari nilai *difference*  $h$  nilai *reduction*  $h'$ , dan nilai *pixel* hasil *mapping*  $b'$ . *Pseudocode* dari fungsi *get\_pixel* ditunjukkan pada *Pseudocode 4.10 Fungsi *get\_pixel* 1* dan *Pseudocode 4.11 Fungsi *get\_pixel* 2*.

```

1.  FUNCTION get_pixel(self, l, h3):
2.      IF self.b <= 1:
3.          IF self.h < 0:
4.              IF self.x MOD 2 = 1:
5.                  lm <- 01
6.                  RETURN lm, 1 - CEIL((h3 + 1) / 2)
7.              ELSE IF self.x MOD 2 = 0:
8.                  lm <- 00
9.                  RETURN lm, 1 - FLOOR(h3 / 2)
10.             ENDIF
11.          ELSE IF self.h >= 0:
12.              IF self.x MOD 2 = 1:
13.                  lm <- 01
14.                  RETURN lm, 1 + FLOOR((h3 - 1) / 2)
15.              ELSE IF self.x MOD 2 = 0:
16.                  lm <- 00
17.                  RETURN lm, 1 + CEIL(h3 / 2).
18.             ENDIF
19.          ENDIF
20.      ELSE IF self.b > 1:
21.          IF self.x MOD 3 = 2:
22.              IF self.b = 3 OR self.b == 4:
23.                  a <- 1
24.              ELSE IF self.b = 2:
25.                  a <- 2
26.              ENDIF
27.          ELSE IF self.x MOD 3 < 2:
28.              IF self.b = 2 OR self.b = 3:
29.                  a <- 0
30.              ELSE IF self.b = 4:
31.                  a <- 1
32.              ENDIF
33.          ENDIF

```

***Pseudocode 4.10 Fungsi get\_pixel 1***



```

34.         IF self.x < 255:
35.             IF self.h < 0:
36.                 IF self.x MOD 3 = 2:
37.                     lm <- 12
38.                     RETURN lm, (1 - CEIL(h3 / 2) - a
39.                 )
40.                 ELSE IF self.x MOD 3 = 1:
41.                     lm <- 11
42.                     RETURN lm, (1 - CEIL((h3 + 1) /
43.                 2) + a)
44.                 ELSE IF self.x MOD 3 = 0:
45.                     lm <- 10
46.                     RETURN lm, (1 - FLOOR(h3 / 2) +
47.                 a)
48.                 ENDIF
49.             ELSE IF self.h >= 0:
50.                 IF self.x MOD 3 = 2:
51.                     lm <- 12
52.                     RETURN lm, (1 + FLOOR(h3 / 2) -
53.                 a)
54.                 ELSE IF self.x MOD 3 = 1:
55.                     lm <- 11
56.                     RETURN lm, (1 + FLOOR((h3 - 1) /
57.                 2) + a)
58.                 ELSE IF self.x MOD 3 = 0:
59.                     lm <- 10
60.                     RETURN lm, (1 + CEIL(h3 / 2) + a
61.                 )
62.                 ENDIF
63.             ENDIF
64.         ELSE IF self.x = 255:
65.             lm <- 1 + STR(self.b2)
66.             RETURN lm, 255
67.         ENDIF
68.     ENDFUNCTION

```

***Pseudocode 4.11 Fungsi get\_pixel 2***

#### 4.2.5.7 Implementasi Fungsi run

Fungsi *run* digunakan untuk menjalankan seluruh fungsi pada kelas setelah fungsi `__init__` dijalankan. Keluaran dari fungsi *run* adalah nilai *pixel* baru  $x'$  dan nilai location map *LM*. *Pseudocode* dari fungsi *get\_difference* ditunjukkan pada *Pseudocode 4.12 Fungsi run*.

```

1. FUNCTION run(self):
2.     l <- self.get_average()
3.     h2 <- self.get_reduction()
4.     h3 <- self.get_embed(h2)
5.     lm, x2 <- self.get_pixel(l, h3)
6.     RETURN lm, x2
7. ENDFUNCTION

```

***Pseudocode 4.12 Fungsi run***

#### 4.2.6 Modul *CalcPSNR*

Modul *CalcPSNR* berisikan fungsi untuk perhitungan *PSNR* dan *MSE*. Perhitungan *PSNR* dan *MSE* menggunakan fungsi *compare\_psnr* dan *compare\_mse* yang diambil dari pustaka *scikit-image* [17]. *Pseudocode* dari perhitungan *PSNR* dan *MSE* ditunjukkan pada *Pseudocode 4.13 Implementasi Perhitungan* .

```

1. FUNCTION calc_psnr(img_inp, img_out):
2.     IF compare_mse(img_inp, img_out) = 0:
3.         psnr_val <- infinity
4.     ELSE:
5.         psnr_val <- compare_psnr(img_inp, img_out)
6.     ENDIF
7.     RETURN psnr_val
8. ENDFUNCTION

```

***Pseudocode 4.13 Implementasi Perhitungan PSNR dan MSE***

#### 4.2.7 Modul *Index*

Modul *Index* berisikan *main function* untuk menghubungkan antar modul dan fungsi lainnya yang berkaitan. Daftar fungsi dalam modul *Index* dapat dilihat pada Tabel 4.3 Daftar Fungsi Modul *Index*. Penjelasan detail fungsi yang ada dalam modul *Index* akan dijelaskan lebih lanjut pada sub poin berikutnya.

**Tabel 4.3 Daftar Fungsi Modul *Index***

<b>Nama Fungsi</b>	<b>Keterangan</b>
<i>base_to_decimal</i>	Mengkonversi <i>payload b</i> dari nilai bilangan basis-n ke dalam nilai desimal.
<i>decimal_to_base</i>	Mengkonversi <i>payload b</i> dari nilai desimal ke dalam nilai bilangan basis-n.
<i>map_lm</i>	Melakukan <i>mapping</i> dari dua digit <i>location map LM</i> menjadi satu digit nilai untuk pengembalian <i>original pixel</i> .
<i>restore_payload</i>	Mengembalikan nilai <i>payload</i> dari <i>location map LM</i> dan <i>stego pixel x'</i> .

#### 4.2.7.1 Implementasi Fungsi *base\_to\_decimal*

Fungsi *base\_to\_decimal* digunakan untuk mengkonversi *payload b* dari nilai bilangan basis-n ke dalam nilai desimal. *Pseudocode* dari perhitungan konversi *payload b* dari nilai bilangan basis-n ke dalam nilai decimal ditunjukkan pada *Pseudocode* 4.14 Fungsi *base\_to\_decimal*.

```

1. FUNCTION base_to_decimal(n, b):
2.     RETURN INT(STR(n), b)
3. ENDFUNCTION

```

***Pseudocode* 4.14 Fungsi *base\_to\_decimal***

#### 4.2.7.2 Implementasi Fungsi *decimal\_to\_base*

Fungsi *decimal\_to\_base* digunakan untuk mengkonversi *payload b* dari nilai desimal ke dalam nilai bilangan basis-*n*. *Pseudocode* dari perhitungan konversi *payload b* dari nilai desimal ke dalam nilai bilangan basis-*n* ditunjukkan pada *Pseudocode 4.15* Fungsi *decimal\_to\_base*.

```

1. FUNCTION decimal_to_base(n, b):
2.     IF n = 0:
3.         RETURN 0
4.     ENDIF
5.     nums <- []
6.     WHILE n:
7.         n, r <- divmod(n, b)
8.         nums.append(STR(r))
9.     ENDWHILE
10.    RETURN ''.join(reversed(nums))
11. ENDFUNCTION

```

*Pseudocode 4.15 Fungsi decimal\_to\_base*

#### 4.2.7.3 Implementasi Fungsi *map\_lm*

Fungsi *map\_lm* digunakan untuk melakukan *mapping* dari dua digit *location map LM* menjadi satu digit nilai yang nilai tersebut akan digunakan untuk proses pengembalian *original pixel*. *Pseudocode* untuk proses *mapping location map* ditunjukkan pada *Pseudocode 4.16* Fungsi *map\_lm*.

```

1. FUNCTION map_lm(lm):
2.     IF lm[0] = 0:
3.         RETURN STR(lm[1])
4.     ELSE IF lm[0] = 1 AND lm_source[1] = 0:
5.         RETURN 2
6.     ELSE IF lm_source[0] = 1 AND lm_source[1] = 1:
7.         RETURN 3
8.     ELSE IF lm_source[0] = 1 AND lm_source[1] = 2:
9.         RETURN 4
10.    ENDIF
11. ENDFUNCTION

```

**Pseudocode 4.16 Fungsi *map\_lm***

#### 4.2.7.4 Implementasi Fungsi *restore\_payload*

Fungsi *restore\_payload* digunakan untuk mengembalikan nilai *payload* dari *location map LM* dan *stego pixel  $x'$* . Pseudocode untuk pengembalian nilai *payload* ditunjukkan pada Pseudocode 4.17 Fungsi *restore\_payload*.

```

1. FUNCTION restore_payload(lm, x1):
2.     IF lm[0] = 0:
3.         IF x1 MOD 2 = 0:
4.             RETURN 0
5.         ELSE IF x1 MOD 2 = 1:
6.             RETURN 1
7.         ENDIF
8.     ELSE IF lm[0] = 1:
9.         IF (x1 < 255 AND x1 MOD 3 = 0) OR (x1 = 255
10. AND lm[1] = 0):
11.             RETURN 2
12.         ELSE
13.             IF (x1 < 255 AND x1 MOD 3 = 1) OR (x1 = 255 AND lm[1]
14. ] = 1):
15.                 RETURN 3
16.             ELSE
17.                 IF (x1 < 255 AND x1 MOD 3 = 2) or (x1 = 255 AND lm[1]
18. ] = 2):
19.                     RETURN 4
20.                 ENDIF
21.             ENDIF
22.         ENDIF
23.     ENDFUNCTION

```

***Pseudocode 4.17 Fungsi restore\_payload***

## **BAB V**

### **HASIL UJI COBA DAN EVALUASI**

Bab ini membahas mengenai uji coba dan evaluasi dari metode yang diusulkan dalam Tugas Akhir.

#### **5.1 Lingkungan Pengujian**

Uji coba dari metode yang diusulkan dalam Tugas Akhir ini dilakukan pada perangkat dengan spesifikasi seperti yang terdapat pada Tabel 5.1 Lingkungan Pengujian.

**Tabel 5.1 Lingkungan Pengujian**

<b>Perangkat</b>	<b>Jenis Perangkat</b>	<b>Spesifikasi</b>
<b>Perangkat Keras</b>	<b>Prosesor</b>	Intel® Core™ i5-8250U CPU @ 1.80 GHz x 4
	<b>Memori</b>	8 GB 2400 MHz DDR4
	<b>Penyimpanan</b>	1 TB Hard Drive
<b>Perangkat Lunak</b>	<b>Sistem Operasi</b>	Windows 10 Home 64-bit
	<b>Perangkat Pengembang</b>	PyCharm Community 2018.1.3

#### **5.2 Data Uji**

Pada sub bab ini akan dijelaskan tentang data uji yang digunakan dalam pengujian metode yang digunakan dalam Tugas Akhir ini. Data uji yang digunakan berupa *cover image* dan daftar *payload*.

##### **5.2.1 Cover Image**

*Cover image* yang digunakan merupakan citra dalam bentuk *grayscale image* berukuran 8-bit dengan dimensi 512x512 *pixel*. *Cover image* diambil dari [23].

### 5.2.2 *Payload*

*Payload* yang digunakan adalah daftar *random binary* yang diambil dari [24].

### 5.3 Skenario Uji Coba

*Payload* yang asalnya dalam bentuk bilangan basis 2 dikonversikan ke bilangan basis 3, 4, dan 5. Uji coba dilakukan dengan menyisipkan *payload* ke dalam tiga *cover image* dan *payload* dengan ukuran 10 kb, 50 kb, dan 100 kb. Jumlah digit *payload* yang dapat disisipkan dan hasil konversi ke bilangan basis 3, 4, dan 5 ditunjukkan pada Tabel 5.2 Jumlah Digit *Payload*.

**Tabel 5.2 Jumlah Digit *Payload***

<b>Ukuran <i>Payload</i> (Basis 2)</b>	<b>Jumlah Digit <i>Payload</i></b>			
	<b>Basis 2</b>	<b>Basis 3</b>	<b>Basis 4</b>	<b>Basis 5</b>
<b>10 kb</b>	10000	6310	5000	4307
<b>50 kb</b>	50000	31547	25000	21534
<b>100 kb</b>	99999	63093	50000	43068

#### 5.3.1 Skenario Uji Coba 1

Pengujian dilakukan dengan menyisipkan *payload* dalam bentuk bilangan basis 2 berukuran 10 kb. Nilai *PSNR* dari hasil penyisipan *payload* berukuran 10 kb dapat dilihat pada Tabel 5.3 Nilai *PSNR* 10 kb *Payload*.



**Tabel 5.3 Nilai *PSNR* 10 kb *Payload***

<i>Cover Image</i>	<b>10 kb <i>Payload</i></b>			
	<b><i>PSNR</i> (dB)</b>			
	<b>Basis 2</b>	<b>Basis 3</b>	<b>Basis 4</b>	<b>Basis 5</b>
<b>Boat</b>	65,35876	64,797256	66,04972	66,050746
<b>Lena</b>	65,401851	64,771955	66,051773	65,907415
<b>Pepper</b>	65,341294	64,909419	66,120053	65,871817

Dari Tabel 5.3 Nilai *PSNR* 10 kb *Payload*, didapatkan bahwa untuk *payload* bentuk basis 2, citra Lena menghasilkan nilai *PSNR* tertinggi dan citra Pepper menghasilkan nilai *PSNR* terendah. Untuk *payload* bentuk basis 3, citra Pepper menghasilkan nilai *PSNR* tertinggi dan citra Lena menghasilkan nilai *PSNR* terendah. Untuk *payload* bentuk basis 4, citra Pepper menghasilkan nilai *PSNR* tertinggi dan citra Boat menghasilkan nilai *PSNR* terendah. Lalu untuk *payload* bentuk basis 5, citra Boat menghasilkan nilai *PSNR* tertinggi dan citra Pepper menghasilkan nilai *PSNR* terendah.

Hasil perhitungan *correlation* antara *cover image* dan *stego image* untuk penyisipan 10 kb *payload* ditunjukkan pada Tabel 5.4 Nilai *Correlation* (*Cover Image* & *Stego Image*) 10 kb *Payload*. Data perhitungan *correlation* antara *cover image* dan *stego image* dengan penyisipan 10 kb *payload* menunjukkan bahwa untuk *payload* dalam bentuk basis 2 sampai dengan basis 5, citra Pepper memiliki nilai *correlation* tertinggi dan citra Boat memiliki nilai *correlation* terendah.

**Tabel 5.4 Nilai *Correlation* (Cover Image & Stego Image) 10 kb Payload**

<i>Cover Image</i>	<b>10 kb Payload</b>			
	<i>Correlation (Cover Image &amp; Stego Image)</i>			
	<b>Basis 2</b>	<b>Basis 3</b>	<b>Basis 4</b>	<b>Basis 5</b>
<b>Boat</b>	0,9999957	0,9999951	0,9999963	0,9999963
<b>Lena</b>	0,9999959	0,9999953	0,9999965	0,9999964
<b>Pepper</b>	0,9999967	0,9999964	0,9999973	0,9999971

Tabel 5.5 Waktu *Embedding* 10 kb *Payload* menunjukkan hasil dari waktu *embedding* untuk *payload* berukuran 10 kb. Data yang didapatkan menunjukkan bahwa semakin besar bentuk basis bilangan pada *payload*, semakin cepat pula waktu untuk proses *embedding*.

**Tabel 5.5 Waktu *Embedding* 10 kb *Payload***

<i>Cover Image</i>	<b>10 kb Payload</b>			
	<b>Waktu <i>Embedding</i> (s)</b>			
	<b>Basis 2</b>	<b>Basis 3</b>	<b>Basis 4</b>	<b>Basis 5</b>
<b>Boat</b>	0,2029998	0,1340001	0,115	0,109
<b>Lena</b>	0,1949999	0,1340001	0,122	0,1040001
<b>Pepper</b>	0,2079999	0,1359999	0,1169999	0,1030002

Tabel 5.6 Waktu *Extracting* 10 kb *Payload* menunjukkan hasil dari waktu *extracting* untuk *payload* berukuran 10 kb. Data yang didapatkan menunjukkan bahwa semakin besar bentuk basis bilangan pada *payload*, semakin cepat pula waktu untuk proses *extracting*.

**Tabel 5.6 Waktu *Extracting* 10 kb Payload**

<i>Cover Image</i>	<b>10 kb Payload</b>			
	<b>Waktu <i>Extracting</i> (s)</b>			
	<b>Basis 2</b>	<b>Basis 3</b>	<b>Basis 4</b>	<b>Basis 5</b>
<b>Boat</b>	0,2359998	0,1569998	0,1439998	0,1270001
<b>Lena</b>	0,2249999	0,1690001	0,142	0,1190002
<b>Pepper</b>	0,233	0,1760001	0,1470001	0,118

Hasil perhitungan *correlation* dari *payload* sebelum disisipkan dan *recovered payload* hasil ekstraksi untuk penyisipan *payload* sebesar 10 kb ditunjukkan pada Tabel 5.7 Nilai *Correlation (Payload & Recovered Payload)* 10 kb *Payload*, sedangkan hasil perhitungan *correlation* pada *cover image* dan *recovered image* hasil ekstraksi untuk penyisipan *payload* sebesar 10 kb ditunjukkan pada Tabel 5.8 Nilai *Correlation (Cover Image & Recovered Image)* 10 kb *Payload*. Kedua tabel tersebut menunjukkan bahwa nilai *correlation* dari *payload* dan *recovered payload* serta nilai *correlation* dari *cover image* dan *recovered image* untuk penyisipan 10 kb *payload* semuanya bernilai 1.

**Tabel 5.7 Nilai *Correlation (Payload & Recovered Payload)* 10 kb Payload**

<i>Cover Image</i>	<b>10 kb Payload</b>			
	<b><i>Correlation (Payload &amp; Recovered Payload)</i></b>			
	<b>Basis 2</b>	<b>Basis 3</b>	<b>Basis 4</b>	<b>Basis 5</b>
<b>Boat</b>	1	1	1	1
<b>Lena</b>	1	1	1	1
<b>Pepper</b>	1	1	1	1

**Tabel 5.8 Nilai *Correlation* (Cover Image & Recovered Image)  
10 kb Payload**

<i>Cover Image</i>	<b>10 kb Payload</b>			
	<i>Correlation (Cover Image &amp; Recovered Image)</i>			
	<b>Basis 2</b>	<b>Basis 3</b>	<b>Basis 4</b>	<b>Basis 5</b>
<b>Boat</b>	1	1	1	1
<b>Lena</b>	1	1	1	1
<b>Pepper</b>	1	1	1	1

### 5.3.2 Skenario Uji Coba 2

Pengujian dilakukan dengan menyisipkan *payload* dalam bentuk bilangan basis 2 berukuran 50 kb. Nilai *PSNR* dari hasil penyisipan *payload* berukuran 50 kb dapat dilihat pada Tabel 5.9 Nilai *PSNR* 50 kb *Payload*.

**Tabel 5.9 Nilai *PSNR* 50 kb *Payload***

<i>Cover Image</i>	<b>50 kb Payload</b>			
	<i>PSNR (dB)</i>			
	<b>Basis 2</b>	<b>Basis 3</b>	<b>Basis 4</b>	<b>Basis 5</b>
<b>Boat</b>	58,342365	57,77065	59,074408	58,952406
<b>Lena</b>	58,333851	57,836565	59,126392	58,894543
<b>Pepper</b>	58,361366	57,805711	59,144557	59,011862

Dari Tabel 5.9 Nilai *PSNR* 50 kb *Payload*, didapatkan bahwa untuk *payload* bentuk basis 2, citra Pepper menghasilkan nilai *PSNR* tertinggi dan citra Lena menghasilkan nilai *PSNR* terendah. Untuk *payload* bentuk basis 3, citra Lena menghasilkan nilai *PSNR* tertinggi dan citra Boat menghasilkan nilai *PSNR* terendah. Untuk *payload* bentuk basis 4, citra Pepper menghasilkan nilai *PSNR* tertinggi dan citra Boat menghasilkan nilai *PSNR* terendah. Lalu untuk *payload* bentuk basis 5, citra Pepper menghasilkan nilai *PSNR* tertinggi dan citra Lena menghasilkan nilai *PSNR* terendah.

Hasil perhitungan *correlation* antara *cover image* dan *stego image* untuk penyisipan 50 kb *payload* ditunjukkan pada Tabel 5.10 Nilai *Correlation (Cover Image & Stego Image)* 50 kb *Payload*. Data perhitungan *correlation* antara *cover image* dan *stego image* dengan penyisipan 50 kb *payload* menunjukkan bahwa untuk *payload* dalam bentuk basis 2 sampai dengan basis 5, citra Pepper memiliki nilai *correlation* tertinggi dan citra Boat memiliki nilai *correlation* terendah.

**Tabel 5.10 Nilai *Correlation (Cover Image & Stego Image)* 50 kb *Payload***

<i>Cover Image</i>	<b>50 kb <i>Payload</i></b>			
	<i>Correlation (Cover Image &amp; Stego Image)</i>			
	<b>Basis 2</b>	<b>Basis 3</b>	<b>Basis 4</b>	<b>Basis 5</b>
<b>Boat</b>	0,9999781	0,9999756	0,9999817	0,999981
<b>Lena</b>	0,9999792	0,999977	0,9999828	0,9999817
<b>Pepper</b>	0,9999837	0,9999817	0,9999865	0,9999859

Tabel 5.11 Waktu *Embedding* 50 kb *Payload* menunjukkan hasil dari waktu *embedding* untuk *payload* berukuran 50 kb. Data yang didapatkan menunjukkan bahwa semakin besar bentuk basis bilangan pada *payload*, semakin cepat pula waktu untuk proses *embedding*.

**Tabel 5.11 Waktu *Embedding* 50 kb *Payload***

<i>Cover Image</i>	<b>50 kb <i>Payload</i></b>			
	<b>Waktu <i>Embedding</i> (s)</b>			
	<b>Basis 2</b>	<b>Basis 3</b>	<b>Basis 4</b>	<b>Basis 5</b>
<b>Boat</b>	0,957	0,7259998	0,5670002	0,4960001
<b>Lena</b>	0,9779999	0,7089999	0,5929999	0,497
<b>Pepper</b>	0,984	0,694	0,586	0,4940002

Tabel 5.12 Waktu *Extracting* 50 kb *Payload* menunjukkan hasil dari waktu *extracting* untuk *payload* berukuran 50 kb. Data

yang didapatkan menunjukkan bahwa semakin besar bentuk basis bilangan pada *payload*, semakin cepat pula waktu untuk proses *extracting*.

**Tabel 5.12 Waktu *Extracting* 50 kb *Payload***

<i>Cover Image</i>	<b>50 kb <i>Payload</i></b>			
	<b>Waktu <i>Extracting</i> (s)</b>			
	<b>Basis 2</b>	<b>Basis 3</b>	<b>Basis 4</b>	<b>Basis 5</b>
<b>Boat</b>	1,1430001	0,8110001	0,6880002	0,5839999
<b>Lena</b>	1,1429999	0,806	0,681	0,5880001
<b>Pepper</b>	1,0910001	0,8069999	0,6570001	0,5929999

Hasil perhitungan *correlation* dari *payload* sebelum disisipkan dan *recovered payload* hasil ekstraksi untuk penyisipan *payload* sebesar 50 kb ditunjukkan pada Tabel 5.13 Nilai *Correlation (Payload & Recovered Payload)* 50 kb *Payload*, sedangkan hasil perhitungan *correlation* pada *cover image* dan *recovered image* hasil ekstraksi untuk penyisipan *payload* sebesar 10 kb ditunjukkan pada Tabel 5.14 Nilai *Correlation (Cover Image & Recovered Image)* 50 kb *Payload*. Kedua tabel tersebut menunjukkan bahwa nilai *correlation* dari *payload* dan *recovered payload* serta nilai *correlation* dari *cover image* dan *recovered image* untuk penyisipan 50 kb *payload* semuanya bernilai 1.

**Tabel 5.13 Nilai *Correlation (Payload & Recovered Payload)* 50 kb *Payload***

<i>Cover Image</i>	<b>50 kb <i>Payload</i></b>			
	<b><i>Correlation (Payload &amp; Recovered Payload)</i></b>			
	<b>Basis 2</b>	<b>Basis 3</b>	<b>Basis 4</b>	<b>Basis 5</b>
<b>Boat</b>	1	1	1	1
<b>Lena</b>	1	1	1	1
<b>Pepper</b>	1	1	1	1

**Tabel 5.14 Nilai *Correlation* (Cover Image & Recovered Image)  
50 kb Payload**

<i>Cover Image</i>	<b>50 kb Payload</b>			
	<b><i>Correlation</i> (Cover Image &amp; Recovered Image)</b>			
	<b>Basis 2</b>	<b>Basis 3</b>	<b>Basis 4</b>	<b>Basis 5</b>
<b>Boat</b>	1	1	1	1
<b>Lena</b>	1	1	1	1
<b>Pepper</b>	1	1	1	1

### 5.3.3 Skenario Uji Coba 3

Pengujian dilakukan dengan menyisipkan *payload* dalam bentuk bilangan basis 2 berukuran 100 kb. Nilai *PSNR* dari hasil penyisipan *payload* berukuran 100 kb dapat dilihat pada Tabel 5.15 Nilai *PSNR* 100 kb *Payload*.

**Tabel 5.15 Nilai *PSNR* 100 kb *Payload***

<i>Cover Image</i>	<b>100 kb Payload</b>			
	<b><i>PSNR</i> (dB)</b>			
	<b>Basis 2</b>	<b>Basis 3</b>	<b>Basis 4</b>	<b>Basis 5</b>
<b>Boat</b>	55,358323	54,827581	56,124017	55,95002
<b>Lena</b>	55,336503	54,867017	56,139596	56,028126
<b>Pepper</b>	55,32034	54,818453	56,158701	56,014773

Dari Tabel 5.15 Nilai *PSNR* 100 kb *Payload*, didapatkan bahwa untuk *payload* bentuk basis 2, citra Boat menghasilkan nilai *PSNR* tertinggi dan citra Pepper menghasilkan nilai *PSNR* terendah. Untuk *payload* bentuk basis 3, citra Lena menghasilkan nilai *PSNR* tertinggi dan citra Pepper menghasilkan nilai *PSNR* terendah. Untuk *payload* bentuk basis 4, citra Pepper menghasilkan nilai *PSNR* tertinggi dan citra Boat menghasilkan nilai *PSNR* terendah. Lalu untuk *payload* bentuk basis 5, citra Lena menghasilkan nilai *PSNR* tertinggi dan citra Boat menghasilkan nilai *PSNR* terendah.

Hasil perhitungan *correlation* antara *cover image* dan *stego image* untuk penyisipan 100 kb *payload* ditunjukkan pada Tabel 5.16 Nilai *Correlation (Cover Image & Stego Image)* 100 kb *Payload*. Data perhitungan *correlation* antara *cover image* dan *stego image* dengan penyisipan 100 kb *payload* menunjukkan bahwa untuk *payload* dalam bentuk basis 2 sampai dengan basis 5, citra Pepper memiliki nilai *correlation* tertinggi dan citra Boat memiliki nilai *correlation* terendah.

**Tabel 5.16 Nilai *Correlation (Cover Image & Stego Image)* 100 kb *Payload***

<i>Cover Image</i>	<b>100 kb <i>Payload</i></b>			
	<i>Correlation (Cover Image &amp; Stego Image)</i>			
	<b>Basis 2</b>	<b>Basis 3</b>	<b>Basis 4</b>	<b>Basis 5</b>
<b>Boat</b>	0,9999566	0,9999528	0,9999642	0,9999621
<b>Lena</b>	0,9999585	0,9999551	0,999966	0,9999646
<b>Pepper</b>	0,9999671	0,9999642	0,9999733	0,999972

Tabel 5.17 Waktu *Embedding* 100 kb *Payload* menunjukkan hasil dari waktu *embedding* untuk *payload* berukuran 100 kb. Data yang didapatkan menunjukkan bahwa semakin besar bentuk basis bilangan pada *payload*, semakin cepat pula waktu untuk proses *embedding*.

**Tabel 5.17 Waktu *Embedding* 100 kb *Payload***

<i>Cover Image</i>	<b>100 kb <i>Payload</i></b>			
	<b>Waktu <i>Embedding</i> (s)</b>			
	<b>Basis 2</b>	<b>Basis 3</b>	<b>Basis 4</b>	<b>Basis 5</b>
<b>Boat</b>	1,891	1,329	1,0940001	0,9749999
<b>Lena</b>	1,924	1,316	1,0939999	0,9659998
<b>Pepper</b>	1,895	1,372	1,0709999	0,967

Tabel 5.18 Waktu *Extracting* 100 kb *Payload* menunjukkan hasil dari waktu *extracting* untuk *payload* berukuran 100 kb. Data



yang didapatkan menunjukkan bahwa semakin besar bentuk basis bilangan pada *payload*, semakin cepat pula waktu untuk proses *extracting*.

**Tabel 5.18 Waktu *Extracting* 100 kb *Payload***

<i>Cover Image</i>	<b>100 kb <i>Payload</i></b>			
	<b>Waktu <i>Extracting</i> (s)</b>			
	<b>Basis 2</b>	<b>Basis 3</b>	<b>Basis 4</b>	<b>Basis 5</b>
<b>Boat</b>	2,1789999	1,582	1,3109999	1,1229999
<b>Lena</b>	2,184	1,5410001	1,312	1,0970001
<b>Pepper</b>	2,2079999	1,569	1,293	1,1069999

Hasil perhitungan *correlation* dari *payload* sebelum disisipkan dan *recovered payload* hasil ekstraksi untuk penyisipan *payload* sebesar 100 kb ditunjukkan pada Tabel 5.19 Nilai *Correlation (Payload & Recovered Payload)* 100 kb *Payload*, sedangkan hasil perhitungan *correlation* pada *cover image* dan *recovered image* hasil ekstraksi untuk penyisipan *payload* sebesar 10 kb ditunjukkan pada Tabel 5.20 Nilai *Correlation (Cover Image & Recovered Image)* 100 kb *Payload*. Kedua tabel tersebut menunjukkan bahwa nilai *correlation* dari *payload* dan *recovered payload* serta nilai *correlation* dari *cover image* dan *recovered image* untuk penyisipan 100 kb *payload* semuanya bernilai 1.

**Tabel 5.19 Nilai *Correlation (Payload & Recovered Payload)* 100 kb *Payload***

<i>Cover Image</i>	<b>100 kb <i>Payload</i></b>			
	<b><i>Correlation (Payload &amp; Recovered Payload)</i></b>			
	<b>Basis 2</b>	<b>Basis 3</b>	<b>Basis 4</b>	<b>Basis 5</b>
<b>Boat</b>	1	1	1	1
<b>Lena</b>	1	1	1	1
<b>Pepper</b>	1	1	1	1

**Tabel 5.20 Nilai *Correlation* (Cover Image & Recovered Image)  
100 kb Payload**

<i>Cover Image</i>	<b>100 kb Payload</b>			
	<i>Correlation (Cover Image &amp; Recovered Image)</i>			
	<b>Basis 2</b>	<b>Basis 3</b>	<b>Basis 4</b>	<b>Basis 5</b>
<b>Boat</b>	1	1	1	1
<b>Lena</b>	1	1	1	1
<b>Pepper</b>	1	1	1	1

#### 5.4 Evaluasi Skenario Uji Coba

Dari seluruh skenario uji coba yang dilakukan, didapatkan bahwa modifikasi dari algoritma *Controlling Expansion* [2] yang dilakukan berhasil mengembalikan seluruh *recovered payload* dan *recovered image*. Hal ini ditunjukkan dengan nilai *correlation* pada *payload* dan *recovered payload* serta *cover image* dan *recovered image* yang bernilai 1 pada semua hasil skenario uji coba.

Modifikasi dari algoritma *Controlling Expansion* [2] yang dilakukan juga dapat meningkatkan kapasitas penyisipan data. Hal ini ditunjukkan dengan semakin menyusutnya jumlah digit *payload* yang disisipkan seiring dengan meningkatnya jumlah basis bilangan *payload* seperti ditunjukkan pada Tabel 5.2 Jumlah Digit *Payload*.

Pada Tabel 5.2 Jumlah Digit *Payload*, jumlah digit *payload* basis 2 dengan ukuran 10 kb adalah 10000 digit, basis 3 adalah 6310 digit, basis 4 adalah 5000 digit, dan basis 5 adalah 4307 digit. Jumlah digit *payload* basis 5 lebih sedikit daripada jumlah digit *payload* basis 4, jumlah digit *payload* basis 4 lebih sedikit daripada jumlah digit *payload* basis 3, dan jumlah digit *payload* basis 3 lebih sedikit daripada jumlah digit *payload* basis 2. Jumlah digit *payload* yang lebih sedikit akan memakan jumlah *pixel* untuk penyisipan yang lebih sedikit pula.

Waktu *embedding* dan *extracting* semakin menurun seiring dengan meningkatnya jumlah basis bilangan *payload*. Hal ini

menunjukkan bahwa peningkatan basis bilangan *payload* yang diterapkan pada modifikasi algoritma *Controlling Expansion* [2] dapat mengoptimalkan waktu *embedding* dan *extracting*. Contoh perbandingan dari waktu *embedding* serta waktu *extracting* adalah pada Tabel 5.5 Waktu *Embedding* 10 kb *Payload* dan Tabel 5.6 Waktu *Extracting* 10 kb *Payload*.

Pada Tabel 5.5 Waktu *Embedding* 10 kb *Payload*, waktu *embedding* untuk *payload* basis 2 pada citra Boat adalah 0,2029998 sekon, basis 3 adalah 0,1340001 sekon, basis 4 adalah 0,115 sekon, dan basis 5 adalah 0,109 sekon. Waktu *embedding* pada *payload* basis 5 lebih sedikit daripada waktu *embedding* pada *payload* basis 4, waktu *embedding* pada *payload* basis 4 lebih sedikit daripada waktu *embedding* pada *payload* basis 3, dan waktu *embedding* pada *payload* basis 3 lebih sedikit daripada waktu *embedding* pada *payload* basis 2.

Pada Tabel 5.6 Waktu *Extracting* 10 kb *Payload*, waktu *extracting* untuk *payload* basis 2 pada citra Boat adalah 0,2359998 sekon, basis 3 adalah 0,1569998 sekon, basis 4 adalah 0,1439998 sekon, dan basis 5 adalah 0,1270001 sekon. Waktu *extracting* pada *payload* basis 5 lebih sedikit daripada waktu *extracting* pada *payload* basis 4, waktu *extracting* pada *payload* basis 4 lebih sedikit daripada waktu *extracting* pada *payload* basis 3, dan waktu *extracting* pada *payload* basis 3 lebih sedikit daripada waktu *extracting* pada *payload* basis 2.

*[Halaman ini sengaja dikosongkan]*

## **BAB VI**

### **KESIMPULAN DAN SARAN**

Pada bab ini akan dibahas mengenai kesimpulan dan saran untuk pengembangan ke depan dari Tugas Akhir yang telah dikerjakan.

#### **6.1 Kesimpulan**

Kesimpulan yang diperoleh dari hasil uji coba yang didapatkan dari modifikasi algoritma *Controlling Expansion* yang diterapkan dalam Tugas Akhir ini adalah:

1. Modifikasi dari algoritma *Controlling Expansion* yang dilakukan dapat digunakan untuk steganografi pada *file* citra.
2. Untuk meningkatkan kapasitas penyisipan data pada steganografi melalui media citra, maka dapat dilakukan dengan meningkatkan basis bilangan *payload* dari bilangan basis 2 sampai dengan bilangan basis 5 yang diterapkan pada modifikasi algoritma *Controlling Expansion*.
3. Untuk mengefisiensikan waktu proses *embedding* dan *extracting* pada steganografi melalui media citra, maka dapat dilakukan dengan meningkatkan basis bilangan *payload* dari bilangan basis 2 sampai dengan bilangan basis 5 yang diterapkan pada modifikasi algoritma *Controlling Expansion* yang juga ditunjukkan dengan meningkatnya waktu proses *embedding* dan *extracting* pada hasil uji coba seiring dengan berkurangnya jumlah *payload* yang didapat dari hasil konversi basis bilangan.

#### **6.2 Saran**

Beberapa saran yang diberikan dari hasil pengujian untuk pengembangan lebih lanjut terkait Tugas Akhir ini adalah:

1. Dilakukan pengembangan untuk meniadakan berkas *location map*.
2. Dilakukan pengembangan untuk penyisipan *payload* dengan basis bilangan di atas basis 5.
3. Dilakukan pengembangan media steganografi menggunakan *color image*.

## DAFTAR PUSTAKA

- [1] D. Artz, "Digital Steganography: Hiding Data within Data," *IEEE Internet Computing*, vol. 5, no. 3, pp. 75-80, 2001.
- [2] D. S. Angreni dan T. Ahmad, "Enhancing DE-based Data Hiding Method by Controlling the Expansion," *2016 4th International Conference on Cyber and IT Service Management*, pp. 1-6, 2016.
- [3] T. Jamil, "Steganography: the art of hiding information in plain sight," *IEEE Potentials*, vol. 18, no. 1, pp. 10-12, 1999.
- [4] J. Tian, "Reversible Data Embedding Using a Difference Expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890-896, 2003.
- [5] D.-C. Lou, M.-C. Hu dan J.-L. Liu, "Multiple Layer Data Hiding Scheme for Medical Images," *Computer Standards & Interfaces*, vol. 31, no. 2, pp. 329-335, Februari 2009.
- [6] Python.org, "What is Python? Executive Summary | Python.org," Python.org, [Daring]. Tersedia: <https://www.python.org/doc/essays/blurb/>. [Diakses 29 Desember 2017].
- [7] Python.org, "General Python FAQ — Python 2.7.14 documentation," Python.org, [Daring]. Tersedia: <https://docs.python.org/2.7/faq/general.html>. [Diakses 30 Desember 2017].
- [8] Python.org, "History and License — Python 2.7.14 documentation," Python.org, [Daring]. Tersedia: <https://docs.python.org/2.7/license.html>. [Diakses 30 Desember 2017].
- [9] Python.org, "python-logo.png," Python.org, [Daring]. Tersedia: [https://www.python.org/static/community\\_logos/python-logo.png](https://www.python.org/static/community_logos/python-logo.png). [Diakses 25 Juni 2018].

- [10] Anaconda, “Anaconda Distribution | Anaconda,” Anaconda, [Daring]. Tersedia: <https://www.anaconda.com/distribution/>. [Diakses 29 Desember 2017].
- [11] Anaconda, “logo-dark.png,” Anaconda, [Daring]. Tersedia: <https://www.anaconda.com/wp-content/themes/anaconda/images/logo-dark.png>. [Diakses 25 Juni 2018].
- [12] OpenCV, “About - OpenCV library,” OpenCV, [Daring]. Tersedia: <https://opencv.org/about.html>. [Diakses 30 Desember 2017].
- [13] OpenCV, “logo.png,” OpenCV, [Daring]. Tersedia: <https://opencv.org/assets/theme/logo.png>. [Diakses 25 Juni 2018].
- [14] P. N. Andono, T. Sujoto dan Muljono, dalam *Pengolahan Citra Digital*, Yogyakarta, Penerbit Andi, 2017, p. 10.
- [15] A. Almohammad dan G. Ghinea, “Stego Image Quality and the Reliability of PSNR,” *2010 2nd International Conference on Image Processing Theory, Tools and Applications*, pp. 215-220, 2010.
- [16] SPSS Tutorials, “Pearson Correlations – Quick Introduction,” SPSS Tutorials, [Daring]. Tersedia: <https://www.spss-tutorials.com/pearson-correlation-coefficient/>. [Diakses 30 Juni 2018].
- [17] scikit-image, “scikit-image: Image processing in Python & — ; scikit-image,” scikit-image, [Daring]. Tersedia: <http://scikit-image.org/>. [Diakses 10 Juli 2018].
- [18] scikit-image, “logo.png,” scikit-image, [Daring]. Tersedia: [http://scikit-image.org/\\_static/img/logo.png](http://scikit-image.org/_static/img/logo.png). [Diakses 10 Juli 2018].
- [19] NumPy, “NumPy — Numpy,” Numpy, [Daring]. Tersedia: <http://www.numpy.org/>. [Diakses 10 Juli 2018].
- [20] GitHub, “numpy/branding/icons at master · numpy/numpy · GitHub,” GitHub, [Daring]. Tersedia:



- <https://github.com/numpy/numpy/tree/master/branding/icons>. [Diakses 10 Juli 2018].
- [21] Python.org, “24.1. Tkinter — Python interface to Tcl/Tk — Python 2.7.15 documentation,” Python.org, [Daring]. Tersedia: <https://docs.python.org/2/library/tkinter.html>. [Diakses 6 Mei 2018].
- [22] PythonProgramming.net, “tkinter-logo.png,” PythonProgramming.net, [Daring]. Tersedia: <https://pythonprogramming.net/static/images/categories/tkinter-logo.png>. [Diakses 25 Juni 2018].
- [23] University of Southern California, “SIPI Image Database - Misc,” University of Southern California, [Daring]. Tersedia: <http://sipi.usc.edu/database/database.php?volume=misc>. [Diakses 23 Oktober 2017].
- [24] Browserling, “Random Binary Number Generator - Create Random Bin Digits - Online - Browserling Web Developer Tools,” Browserling, [Daring]. Tersedia: <https://www.browserling.com/tools/random-bin>. [Diakses 18 Mei 2018].

*[Halaman ini sengaja dikosongkan]*

## LAMPIRAN

### Lampiran A Kumpulan *Cover Image*

Lampiran A berisikan kumpulan *cover image* dalam bentuk *grayscale image* berukuran 8-bit yang digunakan dalam pengujian Tugas Akhir ini. *Cover image* yang diujikan adalah Boat, Lena, dan Pepper yang diambil dari [23].



**Lampiran A 1 *Cover Image* Boat**



**Lampiran A 2 *Cover Image* Lena**



**Lampiran A 3 *Cover Image* Pepper**

**Lampiran B Kumpulan *Stego Image***

Lampiran B berisikan sebagian *stego image* hasil dari skenario pengujian yang dilakukan. *Stego image* yang disajikan adalah hasil pengujian seluruh *cover image* dengan *payload* dalam bentuk *binary* ukuran 10 kb, 50 kb, dan 100 kb yang sudah dikonversikan ke dalam bilangan basis 5.



**Lampiran B 1 *Stego Image* Boat Payload 10 kb Basis 5**



**Lampiran B 2 *Stego Image Lena Payload 10 kb Basis 5***



**Lampiran B 3 *Stego Image Pepper Payload 10 kb Basis 5***



**Lampiran B 4 *Stego Image Boat Payload 50 kb Basis 5***



**Lampiran B 5 *Stego Image Lena Payload 50 kb Basis 5***



**Lampiran B 6 *Stego Image Pepper Payload 50 kb Basis 5***



**Lampiran B 7 *Stego Image Boat Payload 100 kb Basis 5***





**Lampiran B 8 *Stego Image Lena Payload 100 kb Basis 5***



**Lampiran B 9 *Stego Image Pepper Payload 100 kb Basis 5***

*[Halaman ini sengaja dikosongkan]*

## BIODATA PENULIS



Herdito Ibnu Dewangkoro, lahir pada tanggal 23 September 1996 di Klaten, Jawa Tengah. Penulis menempuh program studi S-1 di Departemen Informatika Institut Teknologi Sepuluh Nopember (ITS) Surabaya pada tahun 2014-2018 dengan mengambil bidang Komputasi Berbasis Jaringan (KBJ). Penulis memiliki beberapa pengalaman dalam organisasi seperti menjabat sebagai Anggota Departemen Kesejahteraan Mahasiswa Himpunan Mahasiswa Teknik Computer-Informatika (HMTC) ITS periode 2015-2016, Anggota Forum Silaturahmi Lembaga Dakwah Jurusan (FSLDJ) Lembaga Dakwah Kampus Jamaah Masjid Manarul Ilmi (JMMI) ITS periode 2015-2016, dan Anggota Departemen Syiar Lembaga Dakwah Departemen Keluarga Muslim Informatika (KMI) ITS periode 2015-2017. Penulis juga memiliki pengalaman kepanitiaan dengan menjadi Staff *National Programming Contest (NPC)* Schematics HMTC ITS 2015 dan 2016.